NPS ARCHIVE
1966
KOLLINIATIS, J.

COMPUTERIZED ANALYSIS OF PLATES IN FLEXURE

By LT(GJ) JOHN KOLLINIATIS

Thesis supervisor: Prof.G.M.STURMAN
June 1966

Thesis
K763

COMPUTERIZED ANALYSIS OF PLATES IN FLEXURE BY GRID ANALOGY

by

LT(JG) JOHN KOLLINIATIS R.H.N.

SUBMITTED IN PARTIAL FULFILMENT OF THE
REQUIREMENTS FOR THE DEGREES OF NAVAL
ARCHITECT'S AND MASTER OF SCIENCE IN
MECHANICAL ENGINEERING

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
June 1966

ARCHIVE
[illegible]
ATTIS, J

ABSTRACT

The present thesis will allow Users to analyze rectangular plates with or without a rectangular hole with its edges parallel to the edges of the plate, by using an approximate grid analogy.

The method is completely computerized and the user need only specify properties, geometry, loading and kind of output desirable. The computer programs then will generate a deck of cards, that form a complete STRUDL input. STRUDL itself is an equivalent of STRESS but with much more useful capabilities.

Initially it was intended to make everything an one step process, i.e. to read input in a free-field form, generate all necessary statements of STRUDL, store them on a file and let STRUDL read the file and give the desired output. However due to several delays in the Civil Engineering system Laboratory, STRUDL is not yet available and CDL (Command definition language) is now in the process of experimentation. In consequence the user has to supply his input according to specific format and the linking between this subsystem and STRUDL is not automatic. In other words the process is a two-step one and the user has to take the output of this program and use it as a STRUDL job.

The solution to the flexural plate problem contained herein, is extremely general regarding boundary conditions and loadings so that almost every engineering problem of this nature can be taken care of.

The particular connection of the present job with Naval Architecture is that solutions to bottom and side plating with hydrostatic loads, bulkheads with openings and decks with cargo holes will become readily available with a minimum of human effort and intervention.

ACKNOWLEDGEMENT

The author would like to express his gratitude to Professor G.M.STURMAN of the Civil Engineering Department, who has given so much of his time in supervising this work and for his continuous advice and encouragement.

The author would also like to thank Professor J.H.EVANS of the Department of Naval Architecture and Marine Engineering, for initiating the author's interest in Naval structural Analysis.

The advice of the Staff of the Civil Engineering Department, involved in Computer work, was tremendously valuable to the author and it is fully appreciated.

The author would finally like to express his thanks to Mrs Angelique T.Loukakis for typing this thesis.

A part of this work was initially done at the Computation Center of MIT. However the largest part of it has been done in the Civil Engineering system Laboratory.

TABLE OF CONTENTS

Page

Title Page i
Abstract ii
Acknowledgement iv
Table of contents v
Chapter I INTRODUCTION AND SCOPE 1

Chapter II GRID ANALOGY

2.1 General 4
2.2 Description of grid analogy 6
2.3 Rectangular pattern grid analogy 8
2.4 Limitations and principles of application 16

Chapter III A BRIEF DISCUSSION OF ICES

3.1 General 19
3.2 Some capabilities of Icetran 20

CHAPTER IV COMPUTER PROGRAMS

4.1 Introduction and capabilities 24
4.2 A macroscopic view of the program 26
4.3 Formation of the grid 26
4.4 Boundary conditions 27
4.5 Loading 27
4.6 Output 28

Chapter V PROGRAM DOCUMENTATION
5.1 General 29
5.2 - 5.17 Description of SUBROUTINES 31
Chapter VI DISCUSSION OF RESULTS AND RECOMMENDATIONS FOR FUTURE EXPANSION 68
BIBLIOGRAPHY 71
APPENDIX A FLOW DIAGRAMS 73
APPENDIX B ICETRAN LISTINGS 93
APPENDIX C INPUT AND OUTPUT OF SAMPLE PROBLEM 133
APPENDIX D PROPOSED CDL 148
APPENDIX E A DICTIONARY OF VARIABLES IN COMMON 156
APPENDIX F DIRECTIONS TO USERS 159
I Use of the programs as they are now 159
II Use of the program after CDL has been completed. 165

LIST OF FIGURES AND TABLES

Fig.

2.1	Simple square pattern	7
2.2	Square pattern with additional members	7
2.3	Plate and equivalent grid element	9
2.4	Flexibility matrix	11
4.1	Location and orientation of axes	24
5.1	Scetch illustrating some nomeclature	34
5.2	" " " "	35
C.1	Illustration for sample problem	134
Table 5.1	Values of key parameters and their meaning	63

CHAPTER I

INTRODUCTION AND SCOPE

The development of the large digital computers was a decisive step towards the solution of many engineering problems. Indeed during the past few years people looked back and reexa

but it was impossible to carry it out by hand. It was a [illegible] simple matter to solve all these problems then by a digital computer.

Among all these other problems one is the structural analysis of plates a special case of which is the rectangular plate in flexure. Several scientists and engineers have done a marvellous job in solving special cases of it.

It seems that there would be two distinct approaches in coding the solution of the problem into a computer:

(i) Take each special case and code its exact solution.

(ii) Invent a unified approach applicable in all or most special cases.

The first approach requires a tremendous ammount of work. Besides it could never be called general. For, no matter how many special cases one solves, there will always be some other special case left out.

The second approach seems quite attractive, with the only disadvantage that it is an approximate one. However, in cases where

an exact solution is not known, an approximate one is best than nothing. Besides, very often engineers are interested in getting a first estimate of what is happening in a system.

In the structural analysis of rectangular plates, the grid analogy which will be described in detail in the next chapter, is among the best approximate methods to attack the general problem. In addition since several methods to solve skeletal structures have now been developped, one has only to transform the plate into such a structure, that existing computer formulations can solve.

The goal set forward by the author has been to develop a set of computer programs, able to reduce any rectangular plate to its skeletal equivalent and then solve it as such. Although the sample problem included in the present work is a very simple one, the capabilities of the programs developed are quite general. In fact the choice of the sample problem, made strictly because its solution already exists, is a very unfortunate one in illustrating the application of the method into the more general problem.

The case of a rectangular hole on the plate was included here, due to the interest of Naval Architects on the effect of cargo openings on the strength of ship's decks. Also man-holes on vertical bulkheads forming boundaries of fuell-oil tanks constitute another field of application.

What really makes the method powerful is its generality regarding the boundary conditions and the loading.

Indeed, point supports at any point over the area of the plate, line boundary conditions over all or any part of the edges, concentrated point loads, uniform or linearly distributed line and area loads or any combination of them, are all cases that can be taken care by the programs with a minimum of effort. In addition the ability to get any kind of output desired might be very useful in some cases.

In the following chapters the analogy will be described and the method of its computer application will be explained. In addition directions will be given to the users on how to use the programs.

CHAPTER II

GRID ANALOGY

2.1.- General

Perhaps one could say with complete justification that the general problem of a plate in flexure, has been solved, at least in principle. Indeed the differential equation describing the phenomenon is known for many years and it is only necessary to consider the loading and boundary conditions applicable in each particular problem. However this last step, namely, the solution of the differential equation for a given set of boundary conditions and a given loading, sometimes presents innumerable difficulties from the analytical point of view. Even in the case where a solution can be found in a reasonable number of steps, it is very often expressed in the form of an infinite series, which especially in the case of moments and shears converge rather slowly, a fact that reduces their value.

It is the above difficulty that made engineers search for other, perhaps less accurate but more easily developed solutions of the plate problem. There have been three approaches that yieldapproximate solutions to the plate stresses problem.

(i) Finite differences methods.

(ii) Grid analogies.

(iii) Finite element methods.

Of these, the finite differences approach, depending upon the

boundary conditions, very often requires fictitious extensions of members or consideration of members with fictitious properties. This makes a rather complicated task the coding of the general rectangular plate solution to a digital computer. There is also some suspicion that this method gives results deviating from the exact ones (those that they exist) more than those computed by the other two methods.

The finite element method seems to be quite promising not only for the rectangular plate case but for shells of arbitrary shape. Considerable number of people is working on that method, in the civil Engineering department of M.I.T., but there will be some time before this method becomes available to users.

The grid analogy was selected by the author for the following reasons.

(i) It is rather simple and gives very satisfactory results regarding moment intensities and deflections as it was indicated in (1), (2) and (5).

(ii) It can be rather easily programmed to a digital computer.

(iii) The solution to the equivalent grid can be very easily found by the existing STRESS program or by its new equivalent STRUDL, (7),(8),(9).

(iv) The present work intended primarily for the field of Naval Architecture as well as other professions interested in structural engineering deals almost exclusively with rectangular flat plates or plates with small curvature which can be neglected, without considerable reduction in accuracy.

2.2. Description of grid analogy

The grid analog of a flat plate has been represented differently by different people. Lightfoot in (3) uses a square gridwork consisting of grid beams with flexural rigidity and a set of torsion bars with torsional rigidity with no diagonals. Efimba in (6) in a program written to analyse bridge decks only, uses Tamberg8s method of analysing beam and slab type bridges as torsionally weak grids (10).

The approach taken by Hrennikoff in (2) and Yettram and Husain in (1) is, that the plate can be represented by a plane structure composed of articulated bars, joined by pins normal to the plane of the structure. The pattern has to repeat itself throughout the area of the plate and the outline and external dimensions of the plate should be the same with the corresponding ones of the equivalent structure. Repetition of the pattern does not however mean that the size of the unit has to remain the same. In fact by changing the size of the unit we can satisfy any existingdeviations from the simplest case of a simple rectangular plate, like differences in thickness, holes or differences in material properties.

Hrennikoff set up the problem in a very detailed form and developed the equations that will define the properties of the elements of the unit. The restrictions that he imposes in order to solve for the properties of the members are, that the frame work has to deflect properly:

(i) When subjected to pure bending in the X direction with no bending deformations in the Y direction.

(ii) When in pure bending in the Y direction with no deforma-

tion in the X direction.

(iii) When subjected to pure torsion on the X and Y planes.

These are general restrictions applicable in any case and are useful in examining the suitability of any particular unit to be investigated. The third condition falls out whenever the pattern has the same form in the direction of both axes.

Subsequently Hrenikoff investigated two patterns as shown in fig's (2.1) and (2.2) and proceeded to develop their properties.

Of these two patterns the first one restricts the solution to values of Poisson's ratio of 1/3 only.

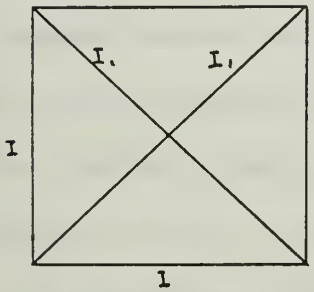


FIG. 2.1

SIMPLE SQUARE PATTERN

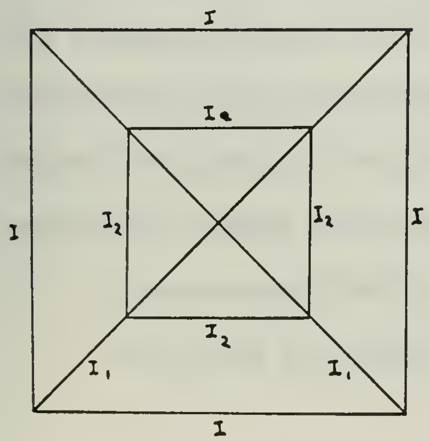


FIG. 2.2 SQUARE PATTERN WITH ADDITIONAL MEMBERS

The second one although valid for any Poisson's ratio has the disadvantage that is a square one. This causes a serious problem in the case that we cannot satisfy the boundary donditions with a square pattern. We also notice that for each square we have 10 slender members, which would make a hand-solution more difficult or a computer solution longer.

A rectangular pattern with diagonals only is much more general and simple. This type of pattern has been very recently investigated by Yettram and Husain in (1). Their pattern

looks to be the best of the existing ones because it is simple, easily developed and considerably general. Due to the fact that the method is relatively new and espesially useful in other kinds of problems (those satisfying the Langrangian equation like thermal stresses in a plate or special cases of fluid flow), it is believed that a relatively detailed discussion ot it must be given. This will be dome in the next chapter.

2.3. Rectangular pattern grid analogy

The idea of the model is that we can substitude a rectangular network consisting of slender members only, in place of a particular plate element. Fig. (2.3a) shows a rectangular element of plating and Fig.(2.3b) its equivalent grid element. The unknown properties of the grid element members are the flexural rigidity of the side beams I_c and I_s , the torsional rigidity of the side beams $\frac{GJ_s}{E}$ and $\frac{GJ_c}{E}$ and the flexural rigidity of the diagonals I_d . That gives 5 unknown properties i.e. I_c , I_s , I_d , $\frac{GJ_s}{E}$ and $\frac{GJ_c}{E}$.

The equations giving the above 5 properties can be developed by equating the rotations at the nodes of the grid element with those of the plating element of equal size. Although the method doesn't give the cross-sectional area of the beams, it gives all the necessary information to make the solution of the grid frame-work possible. Of course the plate and the grid elements must be subject to statically equivalent torques and moments.

Considering first the plating element we examine three cases.

(i) Equal and opposite bending moment intensities on the x sides

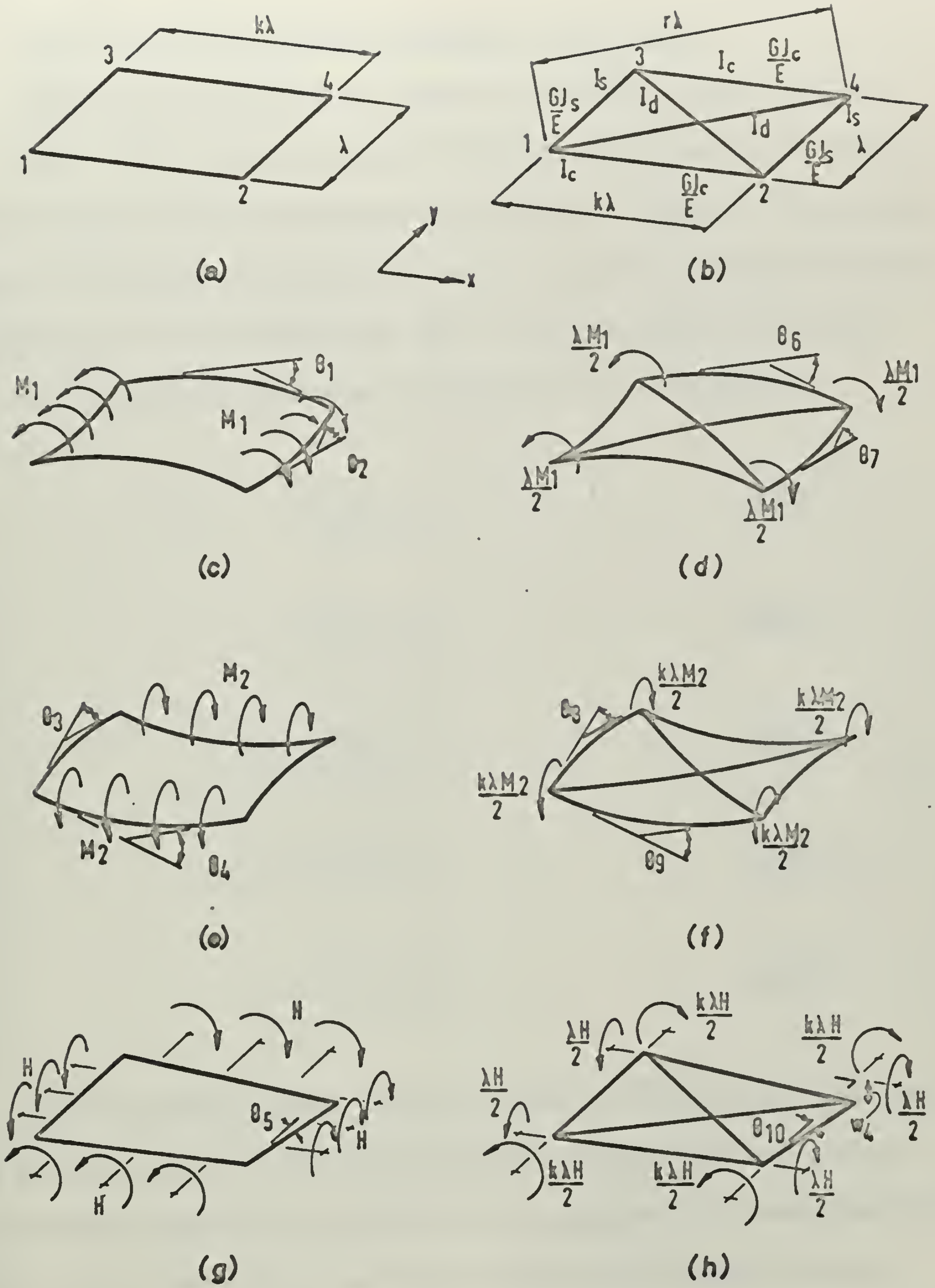


FIG. 2.3 PLATE AND GRID ELEMENTS

(ii) Equal and opposite intensities on y sides.

(iii) Self-equilibrating torque intensities along 4 edges.

Fig's (3d), (3e) and (3g) show clearly the moment and torque distributions and the corresponding angles. The values of the angles in terms of the moment intensities can be found by differentiating the expressions for the deflections, (11). This was done in (1) where they came up with the following expressions for the angles.

$$\theta_1 = \frac{\kappa \lambda M_1}{E(h^3/12)} \qquad (2.1)$$

$$\theta_2 = \frac{\mu \lambda M_1}{E(h^3/12)} \qquad (2.2)$$

$$\theta_3 = \frac{\lambda M_2}{E(h^3/12)} \qquad (2.3)$$

$$\theta_4 = \frac{\mu \kappa \lambda M_2}{E(h^3/12)} \qquad (2.4)$$

$$\theta_5 = \frac{\kappa \lambda H (1+\mu)}{E(h^3/12)} \qquad (2.5)$$

For the grid element loaded as shown in Fig's (3d), (3f) and (3h). Ref (1) gives the flexibility matrix (shown here in Fig 2.4) and concludes that only 2 out of the 12 equations are independent when we have only side bending moments or self-equilibrating torques. By selecting properly these 2 equations in each case we come up with the following expressions for the grid angles.

$$\begin{array}{c|cccccccccccc|c}
M_{1x} & \frac{1}{\lambda}\left(\frac{GJ_c}{k}+4EI_e+\frac{4EI_d}{r^3}\right) & -\frac{4kEI_d}{r^3\lambda} & \frac{6E}{\lambda^2}\left(I_e+\frac{I_d}{r^2}\right) & -\frac{GJ_c}{k\lambda} & 0 & 0 & \frac{2EI_e}{\lambda} & 0 & \frac{6EI_e}{\lambda^2} & \frac{2EI_d}{r^2\lambda} & -\frac{2kEI_d}{r^2\lambda} & \frac{6EI_d}{r^2\lambda^3} & \theta_{1x} \\
M_{1y} & -\frac{4kEI_d}{r^2\lambda} & \frac{1}{\lambda}\left(\frac{4EI_c}{k}+GJ_e+\frac{4k^2EI_d}{r^3}\right) & \frac{6E}{\lambda^3}\left(\frac{I_c}{k^2}+\frac{kI_d}{r^2}\right) & 0 & \frac{2EI_c}{k\lambda} & -\frac{6EI_c}{k^2\lambda^2} & 0 & -\frac{GJ_e}{\lambda} & 0 & -\frac{2kEI_d}{r^2\lambda} & \frac{2k^2EI_d}{r^2\lambda} & -\frac{6kEI_d}{r^3\lambda^3} & \theta_{1y} \\
Q_1 & -\frac{6E}{\lambda}\left(I_e+\frac{I_d}{r^2}\right) & \frac{6E}{\lambda^3}\left(\frac{I_c}{k^2}-\frac{kI_d}{r^2}\right) & \frac{12E}{\lambda^3}\left(\frac{I_c}{k^2}+I_e+\frac{I_d}{r^3}\right) & 0 & \frac{6EI_c}{k^2\lambda^3} & -\frac{13EI_c}{k^2\lambda^2} & -\frac{6EI_e}{\lambda^2} & 0 & -\frac{13EI_e}{\lambda^2} & -\frac{6EI_d}{r^2\lambda^2} & \frac{6kEI_d}{r^3\lambda^2} & -\frac{12EI_d}{r^3\lambda^3} & w_1 \\
M_{2x} & -\frac{GJ_c}{k\lambda} & 0 & 0 & \frac{1}{\lambda}\left(\frac{GJ_c}{k}+4EI_e+\frac{4EI_d}{r^3}\right) & -\frac{4kEI_d}{r^2\lambda} & -\frac{6E}{\lambda^2}\left(I_e-\frac{I_d}{r^3}\right) & \frac{2EI_d}{r^3\lambda} & \frac{2kEI_d}{r^3\lambda} & \frac{6EI_d}{r^3\lambda^2} & \frac{2EI_e}{\lambda} & 0 & \frac{6EI_e}{\lambda^2} & \theta_{2x} \\
M_{2y} & 0 & \frac{3EI_c}{k\lambda} & \frac{6EI_c}{k^2\lambda^3} & \frac{4kEI_d}{r^3\lambda} & \frac{1}{\lambda}\left(\frac{4EI_c}{k}+GJ_e+\frac{4k^2EI_d}{r^3}\right) & -\frac{6E}{\lambda^2}\left(\frac{I_c}{k^2}-\frac{kI_d}{r^2}\right) & \frac{2kEI_d}{r^3\lambda} & \frac{3k^2EI_d}{r^3\lambda} & \frac{6kEI_d}{r^3\lambda^2} & 0 & -\frac{GJ_e}{\lambda} & 0 & \theta_{2y} \\
Q_2 & 0 & -\frac{6EI_c}{k^2\lambda^2} & -\frac{13EI_c}{k^3\lambda^2} & -\frac{6E}{\lambda^2}\left(I_e+\frac{I_d}{r^3}\right) & -\frac{6E}{\lambda^3}\left(\frac{I_c}{k^3}+\frac{kI_d}{r^3}\right) & \frac{12E}{\lambda^3}\left(\frac{I_c}{k^2}+I_e-\frac{I_d}{r^3}\right) & -\frac{6EI_d}{r^3\lambda^3} & -\frac{6kEI_d}{r^3\lambda^2} & -\frac{13EI_d}{r^3\lambda^3} & -\frac{6EI_e}{\lambda^2} & 0 & -\frac{12EI_e}{\lambda^3} & w_2 \\
M_{3x} & \frac{3EI_e}{\lambda} & 0 & -\frac{6EI_e}{\lambda^3} & \frac{2EI_d}{r^3\lambda} & \frac{2kEI_d}{r^3\lambda} & -\frac{6EI_d}{r^2\lambda^2} & \frac{1}{\lambda}\left(\frac{GJ_c}{k}+4EI_e+\frac{4EI_d}{r^3}\right) & \frac{4kEI_d}{r^3\lambda} & \frac{6E}{\lambda^3}\left(I_e+\frac{I_d}{r^3}\right) & -\frac{GJ_c}{k\lambda} & 0 & 0 & \theta_{3x} \\
M_{3y} & 0 & -\frac{GJ_e}{\lambda} & 0 & \frac{2kEI_d}{r^3\lambda} & \frac{2k^3EI_d}{r^3\lambda} & -\frac{6kEI_d}{r^2\lambda^3} & \frac{4kEI_d}{r^3\lambda} & \frac{1}{\lambda}\left(\frac{4EI_c}{k}+GJ_e+\frac{4k^2EI_d}{r^3}\right) & \frac{6E}{\lambda^2}\left(\frac{I_c}{k^2}+\frac{kI_d}{r^3}\right) & 0 & \frac{2EI_c}{k\lambda} & -\frac{6EI_r}{k^3\lambda^2} & \theta_{3y} \\
Q_3 & \frac{6EI_e}{\lambda^2} & 0 & -\frac{13EI_e}{\lambda^2} & \frac{6EI_d}{r^3\lambda^3} & \frac{6kEI_d}{r^3\lambda^3} & -\frac{13kEI_d}{r^3\lambda^2} & \frac{6E}{\lambda^2}\left(I_e-\frac{I_d}{r^3}\right) & \frac{6E}{\lambda^3}\left(\frac{I_c}{k^3}+\frac{kI_d}{r^3}\right) & \frac{12E}{\lambda^2}\left(\frac{I_c}{k^2}+I_e-\frac{I_d}{r^3}\right) & 0 & \frac{6EI_c}{k^2\lambda^2} & -\frac{12EI_c}{k^3\lambda^3} & w_3 \\
M_{4x} & \frac{2EI_d}{r^3\lambda} & -\frac{2kEI_d}{r^3\lambda} & -\frac{6EI_d}{r^2\lambda^2} & \frac{2EI_e}{\lambda} & 0 & -\frac{6EI_e}{\lambda^2} & -\frac{GJ_c}{k\lambda} & 0 & 0 & \frac{1}{\lambda}\left(\frac{GJ_c}{k}+4EI_e+\frac{4EI_d}{r^2}\right) & -\frac{4kEI_d}{r^2\lambda} & \frac{6E}{\lambda^2}\left(I_e+\frac{I_d}{r^2}\right) & \theta_{4x} \\
M_{4y} & -\frac{2kEI_d}{r^2\lambda} & \frac{2k^3EI_d}{r^2\lambda} & \frac{6kEI_d}{r^3\lambda^2} & 0 & -\frac{GJ_e}{\lambda} & 0 & 0 & \frac{2EI_c}{k\lambda} & \frac{6EI_c}{k^2\lambda^3} & -\frac{4kEI_d}{r^3\lambda} & \frac{1}{\lambda}\left(\frac{4EI_c}{k}+GJ_e+\frac{4k^2EI_d}{r^2}\right) & -\frac{6E}{\lambda^2}\left(\frac{I_c}{k^3}+\frac{kI_d}{r^2}\right) & \theta_{4y} \\
Q_4 & \frac{6EI_d}{r^2\lambda^2} & -\frac{6kEI_d}{r^2\lambda^3} & -\frac{12EI_d}{r^2\lambda^2} & \frac{6EI_e}{\lambda^3} & 0 & -\frac{12EI_e}{\lambda^3} & 0 & -\frac{6EI_c}{k^3\lambda^2} & -\frac{12EI_c}{k^3\lambda^3} & \frac{6E}{\lambda^2}\left(I_e+\frac{I_d}{r^2}\right) & -\frac{6E}{\lambda^3}\left(\frac{I_c}{k^3}+\frac{kI_d}{r^2}\right) & \frac{12E}{\lambda^2}\left(\frac{I_c}{k^3}+I_e+\frac{I_d}{r^3}\right) & w_4
\end{array}$$

FIG. 2.4- FLEXIBILITY MATRIX

$$\theta_6 = \frac{M_1 \lambda^2 k}{2E} \cdot \frac{r^3 I_s + I_\alpha}{r^3 I_s I_c + I_\alpha I_c + k^3 I_\alpha I_s} \tag{2.6}$$

$$\theta_7 = \frac{M_1 \lambda^2 k^2}{2E} \cdot \frac{I_\alpha}{r^3 I_s I_c + I_\alpha I_c + k^3 I_\alpha I_s} \tag{2.7}$$

$$\theta_8 = \frac{M_1 \lambda^2 k}{2E} \cdot \frac{r^3 I_c + k^3 I_\alpha}{r^3 I_s I_c + I_\alpha I_c + k^3 I_\alpha I_s} \tag{2.8}$$

$$\theta_9 = \frac{M_2 \lambda^2 k^3}{2E} \cdot \frac{I_\alpha}{r^3 I_s I_c + I_\alpha I_c + k^3 I_\alpha I_s} \tag{2.9}$$

$$\theta_{10} = \frac{H r^3 k \lambda^2}{2E\left[r^3 \left(\frac{G J_s}{E}\right) + 2 k I_\alpha\right]} \tag{2.10}$$

$$W_4 = -\lambda \theta_{10} \tag{2.11}$$

By equating the rotations for the plate and the grid elements we can get the following equations giving the required properties of the grid member. [The method is explained in more detail in (1)].

$$I_s = \frac{(K^2-\mu)\lambda}{2K(1-\mu^2)} \cdot \frac{h^3}{12} \qquad (2.12)$$

$$I_c = \frac{(1-K^2\mu)\lambda}{2(1-\mu^2)} \cdot \frac{h^3}{12} \qquad (2.13)$$

$$I_d = \frac{\mu r^3 \lambda}{2K(1-\mu^2)} \cdot \frac{h^3}{12} \qquad (2.14)$$

$$\frac{GJ_s}{E} = \frac{K(1-3\mu)\lambda}{2(1-\mu^2)} \cdot \frac{h^3}{12} \qquad (2.15)$$

$$\frac{GJ_s}{E} = \frac{(1-3\mu)\lambda}{2(1-\mu^2)} \cdot \frac{h^3}{12} \qquad (2.16)$$

Where k is the ratio of the side beams

μ is the Poisson's ratio

h is the thickness of plating

E is the Young's modulus of elasticity

G is the modulus of rigidity

r is the diagonal to side beam ratio

For future use we will transform the above relations by using:

$$G = \frac{E}{2(1+\mu)} \qquad (2.17)$$

into the following form:

$$I_s = f. \frac{(k^2-\mu)\lambda}{k} \qquad (2.18)$$

$$I_c = f\lambda(1-k^2\mu) \qquad (2.19)$$

$$I_d = f \cdot \frac{2\mu r^3}{k} \qquad (2.20)$$

$$J_c = 2f(1+\mu)(1-3\mu) \qquad (2.21)$$

$$J_s = 2fk(1+\mu)(1-3\mu) \qquad (2.22)$$

where $$f = \frac{h^3}{24(1-\mu^2)} \qquad (2.23)$$

is a constant for a given isotropic plate of constant thickness.

The potentialities of the above equations are immediately apparent. They can be used for the solution of any plating problem, with any boundary conditions and load and any number of holes provided that we can form a grid fine enough so that it will represent a particular plating element, i.e. provided that we will satisfy equations (2.18) through (2.23) at each particular element in the division of the plate. Also by forming a proper grid we can take care of non-isotropic plates.

In the following chapter some limitations and principles of application of the above method will be discussed before the computer approach is explained.

2.4. Limitations and principles of application of Grid method

Expressions (2.18) through (2.23) might give, under special circumstances negative values for the I's or J's. Although Ref.(1) claims that this is not a problem, existing computer programs for the solution of a grid expect positive moments of inertia. This limits the range of applicability of the method in some manner.

The range of the Poisson's ratio for engineering purposes varries between .15 and .3 . Looking carefully into equations (2.18) through (2.13) we conclude that in order to assure positive values of all J's and I's the value of K must be:

$$\sqrt{\mu} \leqslant K \leqslant \sqrt{\frac{1}{\mu}} \qquad (2.24)$$

For $\mu = .3$

$$.55 \leqslant K \leqslant 1.82 \qquad (2.25)$$

For higher values of μ than 1/3, negative properties are unavoidable. However the authors of Ref.(1) state that for good re-

sults the value of k must be between 2 and 0.5.

The computer programs will be such that will automatically form a grid with k's (if more than one) as much close to unity as possible. It is obvious here that the best results are obtained for k=1 but sometimes the boundaries are such that it is quite difficult to have a square pattern.

In the actual application of the method the following points must be taken into account .

(i) The properties found above give the contributions of only one element in the division of the plate. Thus for the side beam we have also to add the properties of the neighbouring elements. Of course this is not true for the diagonals and the members lying along the edges of the plating or along the edges of any existing holes.

(ii) The boundary conditions along the edges of the plating will be considered as applicable on the nodes of the grid at that neighbourhood.

(iii) Any isolated point supports in the interior of the plate are considered applicable on the nearest node.

(d) The loading is considered as acting on the nodes or on the members of the grid.

The exact manner in which the loadings and the boundary conditions are treated will be explained in the description of the particular SUBROUTINES that handle them. Of course most of the assumptions made here require that we have a fine enough grid so that the effect on changing the problem from the original one to another problem will

be minimized.

Having now developed the principles on which the program will operate upon, we are ready to embark in the actual description of the program.

CHAPTER III

A BRIEF DISCUSSION OF ICES

3.1. General

Although this chapter seems fairly obvious to people in the Civil Engineering Department of M.I.T., related to computer work, it is strongly believed that at least a brief description of ICES must be given here for people not familiar with it.

ICES (Integrated Civil Engineering system) is a computer system designed for the efficient solution of a variety of Civil Engineering problems among which one is the structural Analysis Problem.

Although it looks as the system is applicable only to Civil Engineering problems the generality of some SUBSYSTEMS of it makes them quite useful to any other field of Engineering.

The system has been espesially designed in such a way that the user (A user is a person that uses programs written by other people called the programmers, for engineering computational work) communicates with the computer by informing the computer on the nature of the problem and recommending the steps to be taken for the solution of the problem. In addition he doesn't have to provide the input at any particular format. In that way the engineer instructs the computer instead of following instructions imposed by the machine.

ICES has been made to fit into the 360 IBM machine and uses as a programming language, ICETRAN. ICETRAN (ICES, FORTRAN) is a com-

bination of FORTRAN IV plus powerful new capabilities useful in performing engineering work. All ICES programs are supposed to be written in ICETRAN. For the purposes of this work it suffices to say that a program in ICETRAN is precompiled by a special precompiler (the ICETRAN precompiler) into a FORTRAN program which is finally processed by the FORTRAN compiler.

A SUBSYSTEM in ICES is a set of programs that can be used independently or collectively to solve engineering problems. One of the subsystems is STRUDL (STRUCTURAL ANALYSIS & DESIGN) which includes all the capabilities of STRESS plus new ones. In particular the work done herein will hopefully initiate another subsystem called PLATES, which eventually will be able to solve all kinds of problems related to plates.

3.2. Some capabilities of ICETRAN

There have been several capabilities of ICES and ICETRAN that simplify very much this work and in addition make it quite general. These capabilites are:

(a) Interaction between subsystems. Since the approach to this problem has been to reduce it into an equivalent grid and then let STRUDL solve it, the interaction between subsystems is very important. At the time that this work was untertaken this feature was not available and that is why it was decided to make the output of this program a deck of cards that form a complete STRUDL input. However it is a simple matter to modify this program to assure one-step

processing when interaction between subsystems becomes available.

(b) The ICES executive program. It is the ICES executive system that gives the programmer the capability of allowing the user to write his input in an English-like form. The executive process consists of reading the input, finding the command in a dictionary, storing the data in the proper place and executing associated subroutines.

The rules for defining commands and associated procedures are handled by the command Definition Language (CDL) compiler. More information about how to define commands, what the ICEX(ICES EXECUTIVE) input structure looks like etc., may be found in (9), chapter 3.

(c) Dynamic memory allocation

The characteristics of the equivalent Grid might change from one problem to another. We cannot then say that we will arbitrarily set the number of nodes to 500 for examble, since this will depend on the kind of the problem in general and the accuracy that the user desires. That indicates that the arrays to be used throughout the problem are not of a given size but their size varries from problem to problem. As it is well-known in FORTRAN one would have to indicate the size of the different arrays at the beginning of the program. Then if the problem is small (relative to the defined size of arrays) memory storage is waisted. If the problem is large it cannot be executed.

By using the dynamic memory allocation in ICETRAN the programmer can define the size of the arrays during the execution time. This in turn has the effect that at no time the memory is waisted.

In addition to being able to define the size of the arrays

during the execution period another advantage of the dynamic memory allocation is the fact that secondary memory is considered a logical extension of the primary memory. Thus at any time that the primary memory is filled up, the secondary memory automatically is used to store additional information.

The important thing then for the user is that the size of his program is not restricted by the primary memory size. What makes the dynamic memory allocation powerful and nice is the fact that neither the programmer nor the user has to worry about size of arrays, provided that they are dynamic and their size has been defined by an arithmetic expression.

(d) Organization of secondary memory

The last extremely useful feature of ICETRAN, regarding this work is the organization of secondary memory. Since the data generated by the PLATE program will be finally used by STRUDL it looks quite right to write all of it on a file and then let STRUDL read that file and proceed to the solution.

This has not been done at this stage because of the following reasons.

(i) When this work was done STRUDL didn't have the capability to accept information from a file in an internal form.

(ii) The system ROUTINES that can be used to write information on a file cannot do it under a specified FORTRAN type FORMAT, a fact that would allow the writing of information on a FILE in an external form.

(iii) There are statements in FORTRAN that can be used to write information on a FILE under a specified FORMAT statment but the size of the file must be specified in advance (Like the DIMENSION statement) which makes them inadequate for our case.

However when the capabilities (a) and (b) will become available it will be a simple matter to modify the programs in such a way that information will be directly written on a file which STRUDL will be able to read.

This concludes the brief discussion of ICES given in this chapter. The purpose of it, was to give to the non-familiar reader a slight idea, why this work has been done in ICETRAN. A complete description of ICES is given in (9).

CHAPTER IV

COMPUTER PROGRAMS

4.1. Introduction and capabilities

Before a general description of the computer programs is given, it is necessary to give a rather detailed description of the capabilities of the new subsystem.

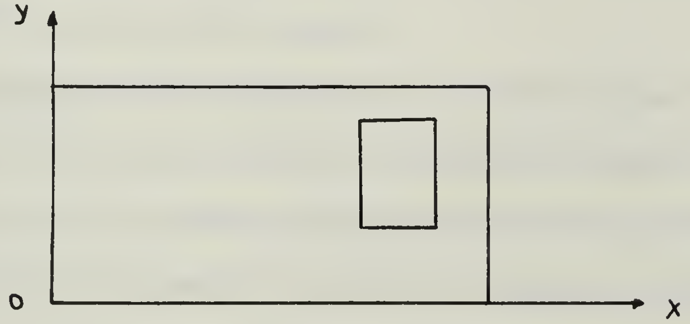


FIG.4.1 - LOCATION AND ORIENTATION OF AXES

Fig. (4.1) indicates the orientation and location of the axes of coordinates relative to the plate. The z - axis is positive upwards. This subsystem will be able to handle constant thickness rectangular plates of constant Young's modulus and Poisson's ratio. The capabilities of the subsystem are:

(i) Number of holes: There can be either no hole at all or a rectangular hole with its edges parallel to the edges of the plate, located anywhere on the surface of the plate.

(ii) Boundary conditions: The boundary conditions that this work can handle can be divided into two categories. The first category

includes line boundary conditions (hinged, free or fixed) arround all or any portion of the edges of the plating. Needless to say that any one of the four edges of the plating might be hinged along one portion of it, free along another and fixed over the rest of it. The second category includes simple point supports over the entire area of the plate except the hole if there is one. No line supports are allowed at any other locations except the four edges. However any line support, in a position other than the edges, can be perfectly well treated as a set of equally spaced point supports.

(iii) Loading: The loading can be of one of three types. A vertical force along the z - axis or a moment the vector of which is parallel to the x or y axis. If the vector of the moment has an orientation other than parallel to the x or y axis it can be treated by considering its components. In any of the above three cases the loading can be a concentrated one acting on a point, a uniform one along a straight line or a linear one along a straight line. Forces not parallel to the z- axis or bending moments having a component parallel to the z - axis connot be treated since this requires a different grid analogy.

(iv) Output: Moments forces or reactions at specified points, along straight lines, or over the entire area of the plating can be requested.

A more detailed description explaining how each case has been taken care of, will be given in the documentation chapter of the programs that perform specific tasks.

4.2. A macroscopic view of the program

The entire work can be logically divided into 4 steps, namely:

(i) Formation of the equivalent grid .

(ii) Reduction of the boundary conditions into equivalent grid supports.

(iii) Reduction of the loads into equivalent grid loads acting on the joints or the members of the grid.

(iv) Reduction of the specified output into an equivalent legitimate grid output.

Each of the 4 steps outlined above is performed by a set of subroutines. Some of the subroutines are equally applicable in both cases where a hole does or doesnot exist. Whenever this is not possible a different subroutine is used for each case. The USER however doesnot have to worry about these details. If he specifies that there or there is not a hole the proper subroutines are executed automatically.

In the next chapters a macroscopic description of each of the above 4 steps will be given.

4.3. Formation of the grid

The first step in the solution of the problem is the generation of the equivalent grid. This step is a combination of geometry and particular boundary conditions in order to produce the proper JOINT COORDINATES, JOINT RELEASES and MEMBER INCIDENCES commands in STRUDL. This part of the job is handled by the following

subroutines:

DIPOHO (Division Points - Hole)

DIPONH (Division Points - No Hole)

GRID (Geometric Formation of Grid - no hole)

MUGRID (Geometric Formation of Multi Grid - there is a hole)

MEINNH (Member Incidences - No hole)

MEINHO (Member Incidences - Hole)

MEMPRO (Member properties)

JOINRE (Joint Releases)

A generalized flow-diagram is given in Fig.4.51. explaining the sequence and interaction of this phase as well as the other three phases.

4.4. Boundary conditions

The boundary conditions of the plate are transformed into an equivalent set of supports for the grid by the same subroutines no matter if there is or there is not a hole.

The specific routines, that do the job here, are:

POBOC (Point Boundary Conditions)

LIBOC (Line Boundary Conditions)

4.5. Loading

There are four subroutines that perform the transformation of the plate loading into an equivalent grid loading. These are:

LINELO (Line Loadings)

POINLO (Point Loadings)

LINE (A logical extension of LINELO to get arround the difficulty of the PRECOMPILER to treat large number of symbols)

AREALO (Area Loading)

4.6. Output

There is only one subroutine (OUT) which in conjuction with POINLO and LINELO is taking care of the output.

CHAPTER V

PROGRAM DOCUMENTATION

5.1. General

This chapter contains adequate information on how the different programs function and how they are linked to each other. This information is necessary especially for other programmers intending to modify a particular subroutine. In fact this chapter is mostly useful to other subprogrammers but it can also be of some help to the users of the program.

Especially for users the chapter on a sample problem and form of data is extremely useful, in guiding them how to write their own input.

In writing the documentation of the SUBROUTINES the author conformed with the recommended methods of (9). Particular attention has been paid on including all the fine points of the programs into either their long description or their flow diagrams. However if any point is not made clear in the documentation of a program the reader is referred to Appendix B which contains all the ICETRAN listings.

Some general points applicable to the entire documentation program are:

(i) All COMMON'S are the same throughout all SUBROUTINES.

(ii) All global dynamic ARRAYS are defined in either DIPONH or DIPOHO and are destroyed by the MAIN PROGRAM.

(iii) All LOCAL DYNAMIC ARRAYS are defined in the particular SUBROUTINE that uses them and destroyed before a return is executed from the subroutine.

(iv) All transmitted dyn. arrays are defined in the CALLING program and destroyed before a return is executed from the calling program.

(v) Since it is impossible at this time to have I/O statements with dynamic array elements, all output is written by first storing the dynamic element into a non-dynamic one and then using I/O statements with this last element.

In the following chapters a description for each subroutine is given. Appendix A contains the flow diagrams, Appendix B the Icetran listings and Appendix E a dictionary of all names in COMMON.

5.2 SUBROUTINE DIPONH

1 - NAME - Subroutine DIPONH

2 - AUTHOR - John Kolliniatis

3 - DATE - May 7, 1966

4 - TYPE - Specific

5 - SHORT DESCRIPTION - This subroutine is used in the case that there is no hole, in order to compute the number of division points along each side for a specified maximum of joints.

6 - LANGUAGE - Icetran

7 - FLOW CHART - As shown in Fig (A5.2) Appendix A.

8 - USAGE - CALL DIPONH with the necessary input values in COMMON.

Input variables are:

VELEN = VELEN1 = Span of plate

HOLEN = HOLEN1 = Width of plate

MANUJO = Desired maximum number of joints.

9 - OUTPUT - Output in COMMON includes:

NUHOJ1 = Number of division points along the x-axis

NUVEJ1 = Number of division points along the y-axis

NJ
NUHOMB
NUVEMB
NURIDI
NULEDI
NUME

} With the same meaning as in subroutine DIPONO

10- LENGTH

Source Program - 60 Icetran instructions

Object Program - 67 Binary cards

11- CALLS - None

12- APPLICATIONS - SUBSYSTEM PLATES

13 - LONG DESCRIPTION - This subroutine computes the division pts along either side of the plate in such a way that the spacing of them will be equal. Since the values found will be fractional it truncates the values computed to the lower integral. Subsequently it examines whether it is possible to increase the division points along only one side so that we approach more to the specified number of division pts.

In addition, since this subroutine is the first one to be executed if there is no hole it defines all the global dynamic arrays.

5.3. SUBROUTINE DIPOHO

1 - NAME - Subroutine DIPOHO

2 - AUTHOR - John Kolliniatis

3 - DATE - April 20, 1966

4 - TYPE - Specific

5 - SHORT DESCRIPTION - This subroutine is used in the case of a rectangular plate with a hole and a specified maximum number of joints for the equivalent grid, in order to find the subdivision points for each of the eight sub-panels that the plate is considered to be consisting of.

6 - LANGUAGE - Icetran

7 - FLOW CHART - As shown in Fig. (A5.3) Appendix A.

8 - USAGE - Call DIPOHO, with the necessary input values in COMMON.

Input variables are:

HOLEN = Length of plate along x-axis

VELEN = Width of plate along y-axis

HOLEN1 = Distance along the x-axis between the center of coordinates and the lower left corner of the hole.

VELEN1 = Distance along the y-axis between the center of coordinates and the lower left corner of the hole

HOLEN2 = Length of hole along the x-axis.

VELEN2 = Width of hole along the y-axis

MANUJO = Desired maximum number of joints for the equivalent grid

9 - OUTPUT - Output in COMMON includes:

NUHOJ1 : Number of division points along the x-axis of sub-panels A, D and F (see Fig. 5.1 and 5.2).

NUHOJ2 : Number of division points along the x-axis of sub-panels B and G.

NUHOJ3 : Number of division points along the x-asis of sub-panels C, E and H.

NUVEJ1 : Number of division points along the y-axis of sub-panels A,B and C.

NUVEJ2 : Number of division points along the y-axis of sub-panels D and E.

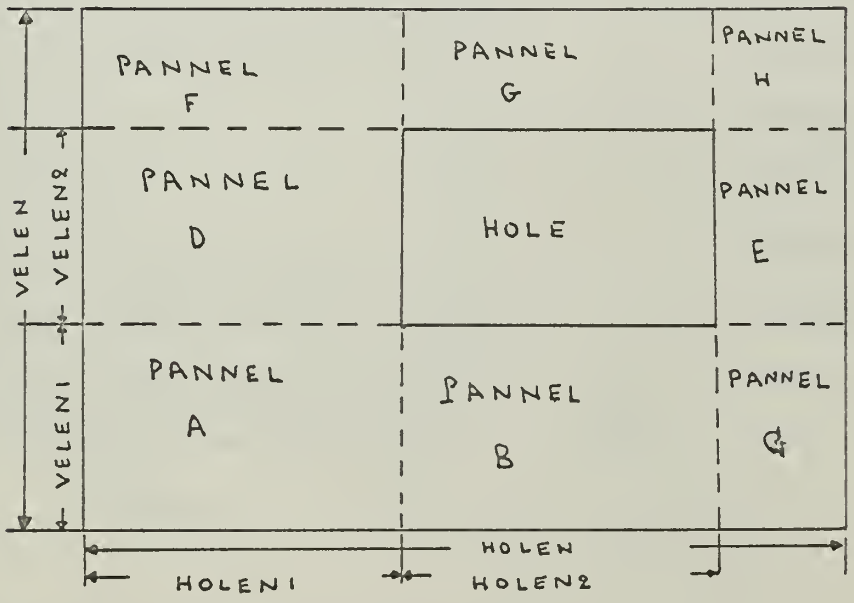


FIG. 5.1 - SCETCH ILLUSTRATING NOMECLATURE

NUVEJ3 : Number of division points along the y-axis of subpanels F, G and H.

NJ : Number of joints.

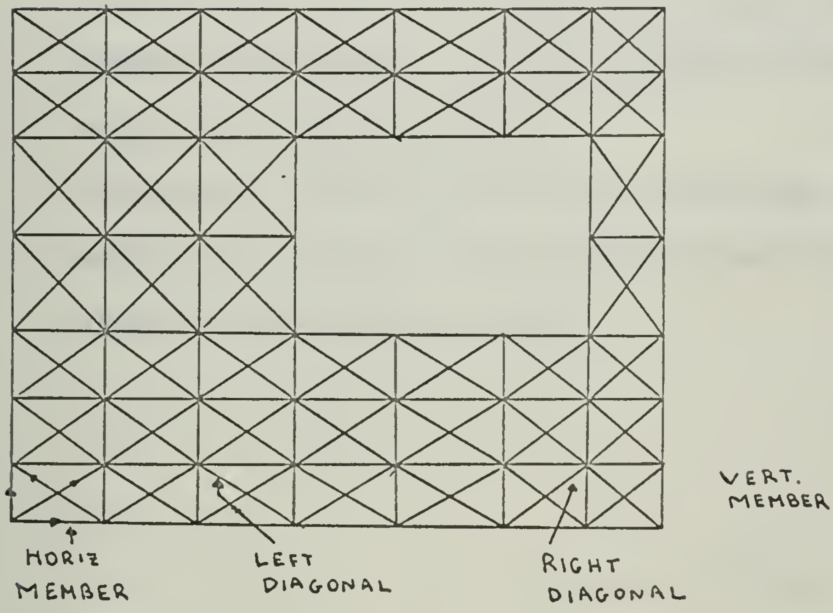
NUHOMB : Number of horizontal members

NUVEMB : Number of vertical members

NURIDI : Number of right - diagonals

NULEDI : Number of left - diagonals

NUME : Total number of members



NUHOJ1 = 4
NUHOJ2 = 4
NUHOJ3 = 2
NUVEJ1 = 4
NUVEJ2 = 3
NUVEJ3 = 3
NUHOMB = 53
NUVEMB = 52
NURIDI = 43
NULEDI = 43
NUME = 191
NJ = 62

FIG.-5.2- SKETCH ILLUSTRATING NOMECLATURE

10 - LENGTH

Source Program - 70 ICETRAN instructions

Object Program - 80 Binary cards

11- CALLS - None

12- APPLICATIONS - SUBSYSTEM PLATES

13- LONG DESCRIPTION - Initially this subroutine examines whether it is possible to select values for the division points of the sub-panels in such a way that any grid element is a square one. Since these results are in fractional form in general a truncation is made to the lowest integral value. In case the hole is very close to one of the sides of the plate, the program overrides the restriction imposed by the USER by his specification of MANUJO, and arranges the division points in such a way that ensures consideration of the hole effect. Of course in this case NJ is larger than MANUJO.

In addition since this subroutine is the first one to be executed in the case of a plate with no holes it is used to define all global dynamic arrays.

5.4. SUBROUTINE GRID

1 - NAME - Subroutine GRID

2 - AUTHOR - John Kolliniatis

3 - DATE - May 7, 1966

4 - TYPE - Specific

5 - SHORT DESCRIPTION - This subroutine is executed after DIPONH in order to form a list of the joints and their coordinates for later use (There is no hole).

6 - LANGUAGE - Icetran

7 - FLOW CHART - As shown in Fig. (A5.5) Appendix A.

8 - USAGE - CALL GRID with the following input variables in COMMON:

$$\left.\begin{array}{l}\text{NJ}\\ \text{NUHOJ1}\\ \text{NUHOJ2}\\ \text{HOLEN}\\ \text{VELEN}\end{array}\right\}\quad \text{with the same meaning as in DIPONH.}$$

9 - OUTPUT - Output in COMMON includes:

HOCOR(I) = x-coordinate of joint I

VECOR(I) = y-coordinate of joint I

NUJO(I) = Number of joint I = I

10- LENGTH -

Source Program -	43	Icetran instructions
Object Program -	30	Binary cards

11- CALLS - None

12- APPLICATIONS - SUBSYSTEM PLATES

13- LONG DESCRIPTION - All that it has to be said here is that the numbering of the joints proceeds from the bottom to the top and from the left to the right. Although the flow diagram is the same as in MUGRID the procedure followed differs appreciably.

5.5. SUBROUTINE MUGRID

1 - NAME - Subroutine MUGRID

2 - AUTHOR - John Kolliniatis

3 - DATE - April 20, 1966

4 - TYPE - Specific

5 - SHORT DESCRIPTION - This subroutine is used in the case of a rectangular plate with a hole in order to number the joints of the equivalent grid and find their horizontal and vertical coordinates.

6 - LANGUAGE - Icetran

7 - FLOW CHART - As shown in Fig. (A55), Appendix A.

8 - USAGE - CALL MUGRID. with the necessary input values in COMMON

Input variables are:

HOLEN1
VELEN1
HOLEN2
VELEN2
NUHOJ1
NUVEJ1
NUHOJ2
NUVEJ2
NUHOJ3
NUVEJ3

having the same meaning as in Subroutine DIPOHO.

HOLEN3 = HOLEN - HOLEN1 - HOLEN2

VELEN3 = VELEN - VELENI - VELEN2

9 - OUTPUT - Output in COMMON includes four dynamic arrays of size NJ where NJ being in COMMON itself, has the same meaning as in DIPOHO. These are:

NUJO(I)= Number of Joint I.

HOCOR(I) = Horizontal coordinate of joint I.

VECOR(I) = Vertical coordinate of joint I.

JORENU(I) - Release number of joint I.

10- LENGTH

Source Program - 107 ICETRAN instructions

Object Program - 45 Machine instructions

11- CALLS - None

12- APPLICATIONS - SUBSYSTEM PLATES

13- LONG DESCRIPTION

This subroutine which is called by the Main program after subroutine DIPOHO has been executed is used in order to find all the nodes of the equivalent grid and their coordinates relative to a system of coordinates located at the left lower corner of the plate. The numbering of the joints is as shown in Fig. (C.1) i.e. it proceeds from the bottom to the top and from the left to the right.

In addition if fills with zero's the elements of the dynamic array JORENU which is supposed to be filled later by another subroutine by characteristic numbers indicating the kind of the joint.

5.6. SUBROUTINE MEINNH

1 - NAME - Subroutine MEINNH

2 - AUTHOR - John Kolliniatis

3 - DATE - May 7, 1966

4 - TYPE - Specific

5 - SHORT DESCRIPTION - This subroutine is called after DIPONH and GRID have been called in order to supply to other subroutines called later information about the number of any member and its minus and plus joint

6 - LANGUAGE - Icetran

7-FLOW CHART - As shown in Fig (A5.7) Appendix A.

8 - USAGE - CALL MEINNH with the following variables and arrays (defined in previous subroutines).

HOCOR
VECOR
NUJO
NUME
NULEDI
NJ
NUHOMB
NUVEMB
NUREDI

9 - OUTPUT - In COMMON includes the following:

NUMEMB(I) = Number of member I = I

NUMIJO(I) = Number of minus joint of member I

NUPLJO(I) = " " plus " " " "

KINDJO(I) = kind of member I

KINDJO(I) = 0 side beam

KINDJO(I) = 1 diagonal

10- LENGTH -

Source Program - 71 Icetran instructions

Object Program - 53 Binary cards

11- CALLS - None

12- APPLICATIONS - SUBSYSTEM PLATES

13- LONG DESCRIPTION - The subroutine examines whether or not a joint is eligible for being a minus joint of a member. If it is the number of the corresponding plus-joint is found and stored together with information indicating the kind of the member.

Although the flow diagram for this subroutine is the same (macroscopically) with the one for MEIMHO, the detailed procedure is different. That was the reason for writing two different subroutines.

5.7. SUBROUTINE MEINHO

1 - NAME - Subroutine MEINHO

2 - AUTHOR - John Kolliniatis

3 - DATE - May 1, 1966

4 - TYPE - Specific

5 - SHORT DESCRIPTION - This subroutine is used in the case of a rectangular plate with a hole. Its task is to assign an identification number on each member. Find its plus and minus joint and store a characteristic number (side beam or diagonal). This information is essential for generating the necessary information for STRUDL (MEMBER INCIDENCES) as well as for supplying data to subroutine MEMPRO for the purpose of computing the properties of the members.

6 - LANGUAGE - Icetran

7 - FLOW CHART - As shown in Fig. (A5.7), Appendix A.

8 - USAGE - CALL MEINHO, with the necessary input values in COMMON. Input variables are:

NUJOJ1
NUHOJ2
NUHOJ3
NUVEJ1
NUVEJ2
NUVEJ3
NUHOMB
NUVEMB

} having the same meaning as in SUBROUTINE DIPOHO.

NURIDI
NUME
NJ

HOCOR VECOR NUJO	having the same meaning as in SUBROUTINE MUGRID.

9 - OUTPUT - Output in COMMON includes four dynamic arrays of size, NUME, namely:

NUMEMB(I) = Number of member I = I.
NUMIJO(I) = Number of minus joint of member I.
NUPLJO(I) = Number of plus joint of member I.
KINDME(I) = Kind of member I.
  KINDME(I) = 0 if member I is a sidebeam
  KINDME(I) = 1 if member I is a diagonal

10- LENGTH

Source Program -	139	ICETRAN instructions
Object Program	85	Machine instructions

11- CALLS - None

12- APPLICATIONS - SUBSYSTEM PLATES

13- LONG DESCRIPTION

The task of this subroutine has been explained in the short description above. For the purposes of explaining how this program functions, the members have been characterized as belonging to one of four teams depending upon the direction of their minus-joint

to plus-joint vector. These four teams are:

(i) vertical members.

(ii) horizontal members.

(iii) right diagonal members.

(iv) left diagonal members.

The numbering of the members proceeds from the bottom to the top and from left to right in each one of the above teams. The sequence of the different teams is as listed above. In other words if there are 20 horizontal members and 30 vertical ones, the bottom right-most, right oriented diagonal member will have the number 51 and so forth.

The method that this subroutine follows is to consider all joints that are legitimate minus joints for a member in sequence (joints on the left edge of a hole are not legitimate minus joints for tight-diagonal members except the upper most one for examble) and compute the corresponding plus-joint. The number of the joint as well as the minus and plus joint numbers and the kind of the member information are stored in the appropriate array elements and the program proceeds with the next member.

5.8. SUBROUTINE POBOC

1 - NAME - Subroutine POBOC

2 - AUTHOR - John Kolliniatis

3 - DATE - May 2, 1966

4 - TYPE - Specific

5 - SHORT DESCRIPTION - This subroutine is used for the purpose of matching the actual supports, with the nearest node of the grid.

6 - LANGUAGE - Icetran

7 - FLOW CHART - As shown in Fig. (A5.8) Appendix A

8 - USAGE - CALL POBOC, with the necessary input values in COMMON.

Input variables are:

HOCOR
VECOR
NUJO
NJ

} same meaning as in previous SUBROUTINES.

EX = Distance along the x-axis between the center of coordinates and the particular point support.

WHY = Distance along the Y-axis between the center of coordinates and the particular point support.

KIND1= Kind of point support:

1 - if support is hinged
2 - if support is fixed

9 - OUTPUT - Output in COMMON includes

KINDJO: An array of size NJ containing numbers with the same

meaning as above for each node, except that zero indicates a regular node (no releases). This information is necessary for subroutine JOINRE in order to produce the proper "JOINT COORDINATES" and "JOINT RELEASES" statements.

10- LENGTH

Source program -	27	ICETRAN instructions
Object program -	30	Machine instructions

11- CALLS - None

12 -APPLICATION - SUBSYSTEM PLATES

13- LONG DESCRIPTION.

For a given point support the distance from each joint is computed and the nearest joint is located. If this joint is not already used as a support, it is considered as such.

If the joint already is a simple support and the point support is fixed, the joint is reclassified as a fixed one.

After the appropriate joint has been located, information is stored in KINDJO regarding its kind.

5.9. SUBROUTINE LIBOC

1 - NAME - Subroutine LIBOC

2 - AUTHOR - John Kolliniatis

3 - DATE - May 2, 1966

4 - TYPE - Specific

5 - SHORT DESCRIPTION - This subroutine takes a line boundary condition and identifies it with a set of grid nodes. Only lines along the 4 edges of the plate, are considered. Any other line boundary condition can be approximated by a set of point supports, but the User has to do it.

6 - LANGUAGE - Icetran

7 - FLOW CHART - As shown in Fig. (A5.9) Appendix A.

8 - USAGE - CALL LIBOC with the necessary input in COMMON (or EQUIVALENCED to other variables in COMMON). Input variables are:

KINOJO
HOCOR
VECOR
NOJO
NJ
} same meaning as in POBOC.

EX1 = Starting (left side) x-coordinate of line boundary condition.

WHY1 = Starting y-coordinate of line boundary condition.

EX2 = Ending (right side) x-coordinate of line boundary condition (EQUIVALENCED to EX).

WHY2 = Ending y-coordinate of line boundary condition.

(EQUIVALENCED to WHY).

KIND2 = Kind of line boundary condition (EQUIVALENCED to KIND1)

KIND2 has the same meaning as KIND1 in POBOC.

9 - OUTPUT - Output in COMMON includes KINDJO with same meaning as in POBOC.

10- LENGTH

Source Program -	61	Icetran instructions
Object Program -	32	Machine instructions

11- CALLS - None

12- APPLICATIONS - SUBSYSTEM PLATES

13- LONG DESCRIPTION - LIBOC is to be executed after POBOC if there are any line boundary conditions. It examines what joints satisfy the equation of the line boundary condition and stores the value of KIND2 into the corresponding elements of KINDJO(I). Of course before it does that it examines whether the line boundary condition coincides in all or in part with one of the 4 edges of the plate. If at any time information is to be stored into a particular element of KINDJO which already contains a number other than zero the larger value if finally kept into the element. In other words if we have two line boundary conditions on the same or different edges touching each other at only one point and one of them is hinged and the other is fixed, the common joint will be considered as fixed.

5.10. SUBROUTINE POINLO

1 - NAME - Subroutine POINLO

2 - AUTHOR - John Kolliniatis

3 - DATE - May 3, 1966

4 - TYPE - Specific

5 - SHORT DESCRIPTION - Initially this subroutine was written in order to take care of the point loads. Later the subroutine has been augmented in order to take also care of the displacements at specified points.

6 - LANGUAGE - Icetran

7 - FLOW CHART - As shown in Fig. (A5.10) Appendix A.

8 - USAGE - CALL POBCC, with the necessary input variables in COMMON. Input variables are:

NUJO
NJ
HOCOR
VECOR
KINDJO
NUJO1

With their usual meaning.

POLOM1 = Point load magnitude.

EXLOAD = x-coord of point of application (EQUIVALENCED TO EX)

WYLOAD = Y-coord of point of application (EQUIVALENCED TO WHY)

NUPOLO = Number of point loads

NLOTY = Type of loading

1 = Z FORCE

2 = X MOMENT

3 = Y MOMENT

KIOUT = A variable that is set by OUT and explained there.

9 - OUTPUT - Output in a punched form consists of a set of "JOINT LOADS" "PRINT DISPLACEMENTS" and "PRINT LOADS" statements.

10- LENGTH

Source Program - 60 ICETRAN instructions

Object Program - 46 Machine instructions

11- CALLS - NOSUJO

12- APPLICATIONS - SUBSYSTEM PLATES

13- LONG DESCRIPTION - The nearest non-support joint is found in the same way as in POBOC. Then according to the value of KIOUT the appropriate statements are punched out. The subroutine itself can handle more than one point loads but in SUBSYSTEM plates it is used for one load each time.

5.11. SUBROUTINE NOSUJO

1 - NAME - Subroutine NOSUJO

2 - AUTHOR - John Kolliniatis

3 - DATE - May 2, 1966

4 - TYPE - Specific

5 - SHORT DESCRIPTION - This subroutine is called after LIBCC and POBCC have been called in order to generate a set of non-support joints

6 - LANGUAGE - Icetran

7 - FLOW CHART - As shown in Fig. (A5.11) Appendix A.

8 - USAGE - CALL NOSUJO with the necessary input variables in COMMON Input variables are:

$$\left.\begin{array}{l}\text{NUJO}\\ \text{HOCOR}\\ \text{VECOR}\\ \text{KINDJO}\end{array}\right\}\quad \text{with the same meaning as in every other subroutine .}$$

NJ

9 - OUTPUT - Output in COMMON includes :

$$\left.\begin{array}{l}\text{NUJO1}\\ \text{HOCOR1}\\ \text{VECOR1}\\ \text{NSIZE}\end{array}\right\}\quad \begin{array}{l}\text{with the same meaning as}\\ \text{NUJO, HOCOR, VECOR, NJ}\\ \text{respectively but for}\\ \text{non-support joints only.}\end{array}$$

10- LENGTH

Source Program -	21	Icetran instructions
Object Program -	25	Machine instructions

11- CALLS - None

12- APPLICATIONS - Subroutine PCINLO OF SUBSYSTEM PLATES.

13- The value of KINDJO is examined for each joint. If it is zero the joint is stored as a non-support joint.

5.12. SUBROUTINE JOINRE

1 - NAME - Subroutine JOINRE

2 - AUTHOR - John Kolliniatis

3 - DATE - May 2, 1966

4 - TYPE - Specific

5 - SHORT DESCRIPTION - Subroutine JOINRE is used in order to write all the introductory STRUDL statements, the JOINT COORDINATES table, the JOINT RELEASES statement and the MEMBER INCIDENCES table.

6 - LANGUAGE - Icetran

7 - FLOW CHART - As shown in Fig (A5.12) Appendix A.

8 - USAGE - CALL JOINRE, with the necessary input variables in COMMON. Input variables are:

NUJO
HOCOR
VECOR
NJ
KINDJO

} with the same meaning as in all previous SUBROUTINES.

JURENU = Joint Release number

9 - OUTPUT - Output in COMMON includes all the statements mentioned in 5 above, in a punched form.

10- LENGTH

Source Program -	89	ICETRAN instructions
Object Program -	68	Binary cards

11- CALLS - None

12- APPLICATIONS - SUBSYSTEM PLATES

13- LONG DESCRIPTION:- The introductory statements are punched first. Then according to the value of KINDJC a table of JOINT COORDINATES is punched with the proper EXISTENCE DATA, support, if joint is a support. Subsequently according to the value of JORENU(J) set by LIBOC the proper JOINT RELEASES statements are punched.

In particular if

JORENU(J) = 0 ; Joint J, releases Moments X,Y
JORENU(J) = 1 ; Joint J releases Moment Y
JORENU(J) = 2 ; Joint J releases Moment X
JORENU(J) = 3 ; Joint J releases Moment X,Y

Finally the "MEMBER INCIDENCES" table is punched out.

5.13. SUBROUTINE MEMPRO

1 - NAME - Subroutine MEMPRO

2 - AUTHOR - John Kolliniatis

3 - DATE - May 2, 1966

4 - TYPE - Specific

5 - SHORT DESCRIPTION - This subroutine computes and punches STRUDL statements for the properties of the members of the grid.

6 - LANGUAGE - Icetran

7 - FLOW CHART - As shown in Fig. (A5.13) Appendix A.

8 - USAGE - CALL MEMPRO, with the necessary input variables in COMMON. Input variables are:

NUHOMB NUVEMB NURIDI NUME NUMIJO NUPLJO NUMEMB HOCOR VECOR	Have same meaning as in any other subroutine.

PRATIO = Poisson's ratio

THICK = Plating thickness

9 - OUTPUT - The output conists of a set of acceptable STRUDL statements that describe the properties of the various members.

10- LENGTH

Source Program - 83 ICETRAN instructions

Object Program - 92 Binary cards

11- CALLS - None

12- APPLICATIONS - SUBSYSTEM PLATES

13- LONG DESCRIPTION - The kind of each member is examined (side beam or diagonal) and then after computing the length of it and all the necessary dimensionless ratio's, the properties of the member are computed by using equations 2.18 through 2.23 of chapter II. Finally all the above information is punched out, forming part of the STRUDL input.

5.14. SUBROUTINE LINELO

1 - NAME - Subroutine LINELO

2 - AUTHOR - John Kolliniatis

3 - DATE - May 3, 1966

4 - TYPE - Specific

5 - SHORT DESCRIPTION - Subroutine LINE is used in order to find the appropriate point (nodes or points along the side members), where concentrated loads have to be applied in order to approximate a specified line load. It also handles cases of output along a straight line.

6 - LANGUAGE - Icetran

7 - FLOW CHART - As shown in Fig. (A5.14) Appendix A.

8 - USAGE CALL - LINELO with the necessary input variables in COMMON. Input variables are:

HOCOR, VECOR, KINDJO, NUMEMB, NUMIJO, NUPLJO, NUVEMB, NUHOMB, NJ with the same meaning as in previous subroutines. In addition:

XA = Start (left) x-coordinate (EQUIVALENCED to EX)

YA = " " y-coordinate (EQUIVALENCED to WHY)

XB = end (right) x-coordinate (EQUIVALENCED to EX2)

YB = " " y-coordinate (EQUIVALENCED to WHY2)

STALD = Start loading Magnitude (EQUIVALENCED to POLOM1)

ENDLD = End loading Magnitude

NLCTY = Same meaning as in POINLO

LTYPE = 0 if uniform load

1 if linear load

9 - OUTPUT - Output in a punched form includes STRUDL statements that will be explained in the chapter for LINE since this subroutine performs all I/O operations for LINELO

10- LENGTH

Source Program -	230	Icetran instructions
Object Program -	281	Binary cards

11- CALLS - LINE

12- APPLICATIONS - SUBSYSTEM PLATES

13- LONG DESCRIPTION

If the line load does not coincide with a set of members, then the intersections of the line load projection with all members intersected by it are found. The number of the intersected member and the local and global coordinates of it are stored for later use by LINE. Special consideration is also given to the case where the line passes thru a joint or a small area arround it in order to be considered as a JOINT LOAD instead of a LINE LOAD.

In case that the line coincides with a set of members then the number of the coinciding (with the line) members as well as local and global coordinates of the start and end loading point for each member, are stored for later use by LINE.

In case that this subroutine is used for a line output certain parts of it are by-passed.

Finally subroutine LINE is called which in addition to other tasks performs also all punching necessary for LINELO.

5.15. SUBROUTINE LINE

1 - NAME - Subroutine LINE

2 - AUTHOR - John Kolliniatis

3 - DATE - May 3, 1966

4 - TYPE - Specific

5 - SHORT DESCRIPTION - Subroutine LINE is a logical extension of LINELO separatedfrom it in order to get arround the difficulty of the precompiler to accept a large number of symbols. It is called by LINELO in order to compute the different grid loads that are equivalent to the plate line loads specified. In addition it performs all the I/O functions for LINELO for both the case of line loads and line outputs.

6 - LANGUAGE - Icetran

7 - FLOW CHART - As shown in Fig. (A5.15) Appendix A.

8 - USAGE - CALL LINE (KINDLO, XGLLD, YGLLD, NULOME, XLOCAL, YLOCAL)

In addition all the input in COMMON in LINELO must also be in COMMON in LINE plus the variable KIOUT.

The values transmitted between LINELO and LINE as arguments have different meanings according to the value of the variable KINDLO. KINDLO itself is an array of size NULDS which contains numbers 0,1,2, selected as it was indicated in the flow diagram for LINELO. (See Appendix E for meaning of NULDS)

In particular:

(i) If KINDLO(J) = 0 (Member concentrated load)

XGLLD = Global x coordinate of concentrated member load

YGLLD = Global y coordinate of " " "

NULOME = Number of loaded member

XLOCAL = Local x coordinate of concentrated member load

YLOCAL = Local y " " " " "

(ii) If KINDLO(J) = 1 (joint load)

$\left.\begin{array}{l}\text{XGLLD}\\ \text{YGLLD}\end{array}\right\}$ = Global coordinates of loaded joint

NULOME = Number of loaded joint

(iii) If KINDLO(J) = 2 (Member uniform or linear load)

XGLLD = Global start of member line load

YGLLD = " end " " " "

XLOCAL = Local start of member line load

YLOCAL = Local end " " " "

NULOME = Number of loaded member

In this case it is necessary to say that XGLLD, YGLLD, XLOCAL, YLOCAL, all refer to the changing coordinate, since in this case either the x or y coordinate remains unchanged.

9 - OUTPUT - Output in a punched form depends upon the values of different key parameters. Table (5.15.1) gives the different combinations of these parameters

10- LENGTH -

Source Program - 269 Icetran instructions

Object Program - 250 Binary cards.

11- CALLS - None

12- APPLICATIONS - SUBSUSTEM PLATES

13- LONG DESCRIPTION - If KINDLO(J) = 2 the list of the points that will be used to approximate the loading is shorted according to the distance from the left end (or lower end if load line is parallel to y axis).

If KINDLO(J) = 2 this step is unnecessary and it is skipped. Each one of the above points takes so much of the load as it is applied between the middle points of the segments joining it with its neighbouring points. Of course this is not the case when KINDLO(J) = 2, for all J's. In this case each member is loaded with the exact linear or uniform load acting over it. Subsequently the necessary output is punched according to the values of the control variables as in table (5.15.1).

KIOUT	KINDLO	NLOTY	LTYPE	STATEMENT PUNCHED OUT	JOINT OR MEMBER LOAD
1	0	1	0 or 1	J* FORCE Z CONCENTRATED	MEMBER
1	0	2	0 or 1	J MOMENT X —— " ——	— " —
1	0	3	— " —	J MOMENT Y —— // ——	— // —
1	1	1	— " —	J FORCE Z —— // ——	JOINT
1	1	2	— // —	J MOMENT X —— // ——	— // —
1	1	3	— " —	J MOMENT Y —— // ——	— // —
1	2	1	0	J FORCE Z UNIFORM K**	MEMBER
1	2	1	1	—— // —— LINEAR — // —	— // —
1	2	2	0	J MOMENT X UNIFORM K	— // —
1	2	2	1	—— " —— LINEAR — " —	— // —
1	2	3	0	J MOMENT Y UNIFORM K	— // —
1	2	3	1	—— // —— LINEAR — " —	— // —
4	Anything	Anything	Anything	LIST LOADS J	JOINT
7	— " —	— " —	— // —	LIST DISPLACEMENTS J	— // —

TABLE 5.1

VALUES OF KEY PARAMETERS AND THEIR MEANING

5.16. SUBROUTINE AREALO

1 - NAME - Subroutine AREALO

2 - AUTHOR - John Kolliniatis

3 - DATE - May 3, 1966

4 - TYPE - Specific

5 - SHORT DESCRIPTION - Subroutine AREALO is used in order to treat area loads. The method is used to divide the loading among the members covered by the area load and then call SUBROUTINE LINELO. Only loads with perimeter rectangular, with the edges parallel to the edges of the plate can be treated.

6 - LANGUAGE - Icetran

7 - FLOW CHART - As shown in Fig. (A5.16) Appendix A.

8 - USAGE - CALL AREALO with the necessary input variables in COMMON. Input variables are:

HOCOR, VECOR, NUMIJO, NUPLJO, NUVEMB, NUHOMB, NLOTY, with their usual meaning.

XA = x-coordinate of lower left corner of area load (EQUIVALENCED to EX)

YA = y-coordinate of lower left corner of area load (EQUIVALENCED to WHY)

XB = x-coordinate of upper right corner of area load (EQUIVALENCED to EX2)

YB = y-coordinate of upper right corner of area load (EQUIVALENCED to WHY2)

LDMSTA = Starting loading magnitude (at lower or left edge)

LTYPE = A variable that indicates the kind of the load; in particular:

LTYPE = 0	Uniform load
LTYPE = 1	Load linear along the x-axis
LTYPE = 2	" " " y-axis

10- LENGTH -

Source Program -	130	Icetran instructions
Object Program -	109	Binary cards

11- CALLS - Subroutine LINELO

12- APPLICATIONS - SUBSYSTEM PLATES

13- LONG DESCRIPTION - The members covered by the load are first located. If LTYPE is 0 or 1 the load is distributed among horizontal members. If LTYPE = 2 the load is distributed among vertical members. Then subroutine LINELO is called in order to punch the necessary output statements.

5.17. SUBROUTINE OUT

1 - NAME - Subroutine OUT

2 - AUTHOR - John Kolliniatis

3 - DATE - May 3, 1966

4 - TYPE - Specific

5 - SHORT DESCRIPTION - This subroutine examines the value of KIOUT, which is set by the user's data or COMMAND (if CDL is used) and according to its value the necessary steps are performed by OUT or other subroutines in order to produce the specified OUTPUT.

6 - LANGUAGE - Icetran

7 - FLOW CHART - As shown in Fig. (A5.17) Appendix A.

8 - USAGE - Call POBOC with the necessary input variables in COMMON. Input variables are:

KIOUT indicates desired output. In particular if:

KIOUT = 2 - LIST REACTIONS

KIOUT = 3 - LIST POINT LOADS XA.. YA..

KIOUT = 4 - LIST LINE LOADS XA... YA... XB... YB

KIOUT = 5 - LIST LOADS

KIOUT = 6 - LIST POINT DISPLACEMENTS XA... YA

KIOYT = 7 - LIST LINE DISPLACEMENTS XA... YA... XB... YB

KIOUT = 8 - LIST DISPLACEMENTS

OUTXA = x and y coordinates of point

OUTYA = where output is requested

OUTXA
OUTYA
OUTXB
OUTYB
} x and y, start and end coordinates of line along which output is desired

9 - OUTPUT - The output is in a form of punched STRUDL statements that take care of any desired collective or selective output that the USER desires.

10- LENGTH

Source Program - 41 Icetran instructions
Object program - 27 Binary cards

11- CALLS - LINELO, POINLO

12- APPLICATIONS - SUBSYSTEM PLATES

13- LONG DESCRIPTION - Flow diagram and short description indicate completely how this subroutine functions.

CHAPTER VI

DISCUSSION OF RESULTS AND RECOMMENDATIONS FOR FURTHER EXPANSION

Several problems have been runned, for the purpose of debbuging individual subroutines and examining the interaction of them. The results have been excellent regarding the generated output. Only the input and output for a very simple problem was included in Appendix F. It covers the case of a square plate with two clamped and two hinged edges under linear, lateral, area load. It has been selected since is one of the solved problems in (11).

There is nothing that can be said on the overall accuracy of the method since the non-availability of STRUDL prevented the author from solving the problem completely. However judging from the results of (1), where a grid has been generated manually, the larger deviation from values measured by experiment are of the order of 15%.

After the displacements have been found the moments and shears can also be evaluated (more correctly) from the following relations given in (2).

$$M_x = -D\left(\frac{\partial^2 w}{\partial x^2} + \mu \frac{\partial^2 w}{\partial y^2}\right) \qquad (6.1)$$

$$M_y = -D\left(\frac{\partial^2 w}{\partial y^2} + \mu \frac{\partial^2 w}{\partial x^2}\right) \qquad (6.2)$$

$$Q_x = -D\left(\frac{\partial^3 w}{\partial x^3} + \frac{\partial^3 w}{\partial x \partial y^2}\right) \qquad (6.3)$$

$$Q_y = -D\left(\frac{\partial^3 w}{\partial x^2 \partial y} + \frac{\partial^3 w}{\partial y^3}\right) \qquad (6.4)$$

where

w = lateral displacement

$$D = \frac{E h^3}{12(1-\mu^2)} \qquad (6.5)$$

Before we state some recommendations for future expansion of this work, there are several other points that must be said regarding its completion. First the output from this program has to be tested in STRUDL when it becomes available. Second the CDL program (a first draft of which is shown in Appendix D) has to be compiled, tested with dummy subroutines and the subroutines of this program. This will make the program for rectangular plates in flexure an integrated one.

In the very restricted category of rectangular plates the

bending problem is not the only problem. There also axial loads, instability or combinations of them. A further expansion of this work, therefore, would be to write similar programs for these cases and then put everything together. It would really be a great contribution to engineering to be able to analyze any plate with any combination of lateral and axial loads in a matter of minutes.

BIBLIOGRAPHY

1.- Yettran A. and Husain H. "Grid Framework method for plates in flexure". Journal of the engineering mechanics division, ASCE, volume 91, June 1965.

2.- Hrennikoff P.A. " Plane stress and bending of plates by method of articulated Framework " . Sc, D. Thesis, Civil Engineering Dept. MIT, Cambridge Mass., May 1940.

3.- Lighfoot E. " A grid framework analogy for laterally loaded plates " by E. Lighfoot, International Journal of Mechanical Sciences, Pergamon Press, Oxford, England. Vol.6, June 65.

4.- Lighfoot, E. " Discussion of A grid framework analogy for laterally loaded plates " by E.Lighfoot, International Journal of Mechanical Sciences, Pergamon Press, Oxford, England. Vol.7, February 1965.

5.- Yettram, Alan L. and Husain H. " The representation of a plate in flexure by a Grid of Orthogonally connected Beams " International Journal of Mechanical Sciences, Pergamon Press, Oxford, England. Vol.7, April 1965.

6.- Efimba, R.E. " Analysis of a bridge deck as a plane grid" MIT Report, Course I, May 1965.

7.- Fenves S.J., et al. SRESS. A User's Manual Civil Engineering Department MIT., The MIT Press, Cambridge, Mass. 1964.

8.- Fenves, S.J.; Logcher, R.D.; Mauch, S.P. STRESS. A reference manual, Civil Eng. Dep., MIT., The MIT Press, Cambridge, MASS. 1965

9.- ICES Architecture Groop, ICES: Programmers guide, MIT report, Course I, November 1965.

10.- Tamberg, K.G. " The Analysis of Right Multi Girder, Simple-Span Bridge Decks as Equivalent Torsionally - Weak Grid Systems " , Report 102, Dept. of Highways, Ontario, Canada, January 1965.

11.- Timoshenko "Theory of plates and shells " , Mc Graw - hill Company, New York, 1940.

APPENDIX A

FLOW DIAGRAMS

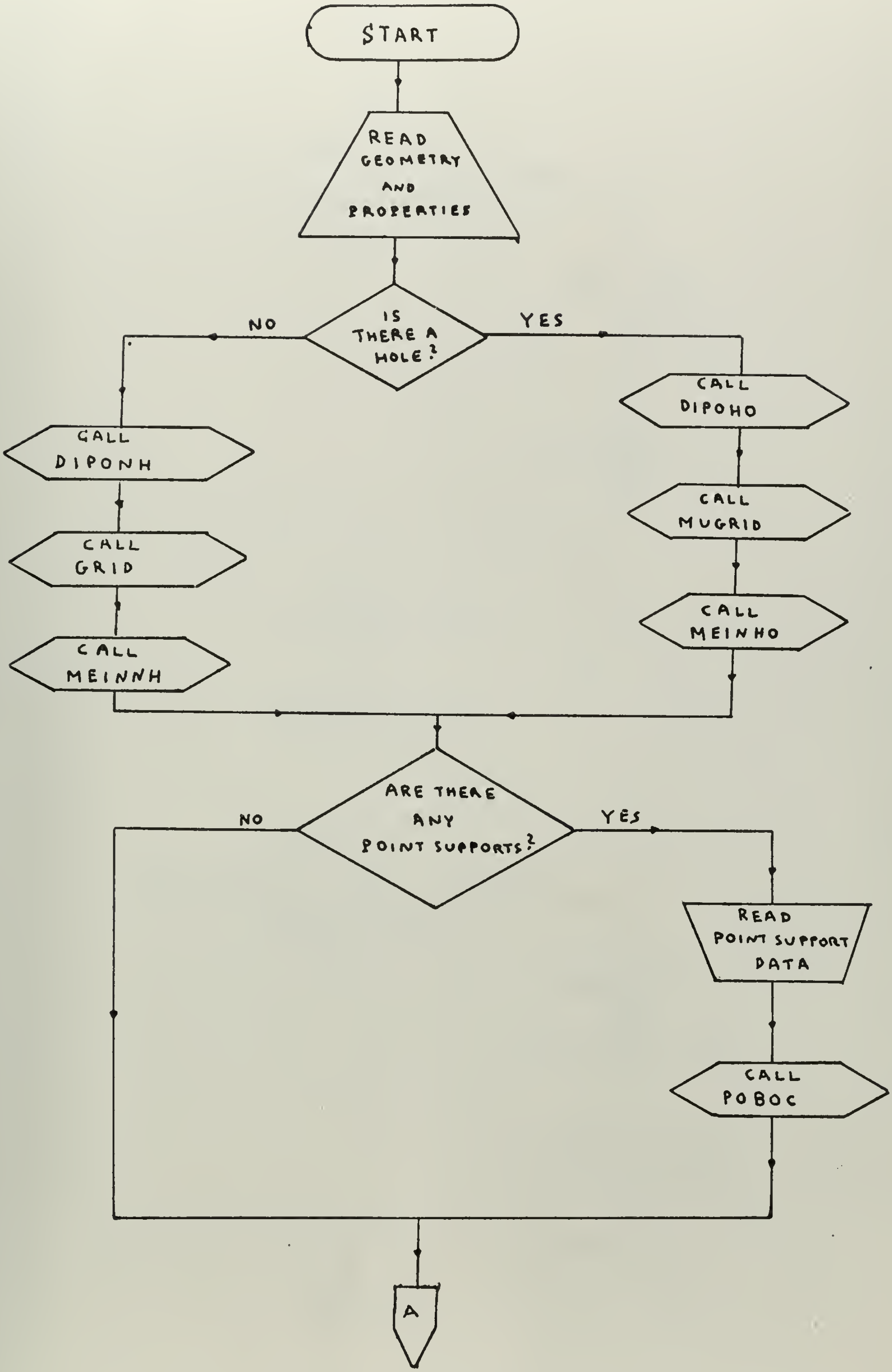


Fig. (A5.1)

MAIN PROGRAM

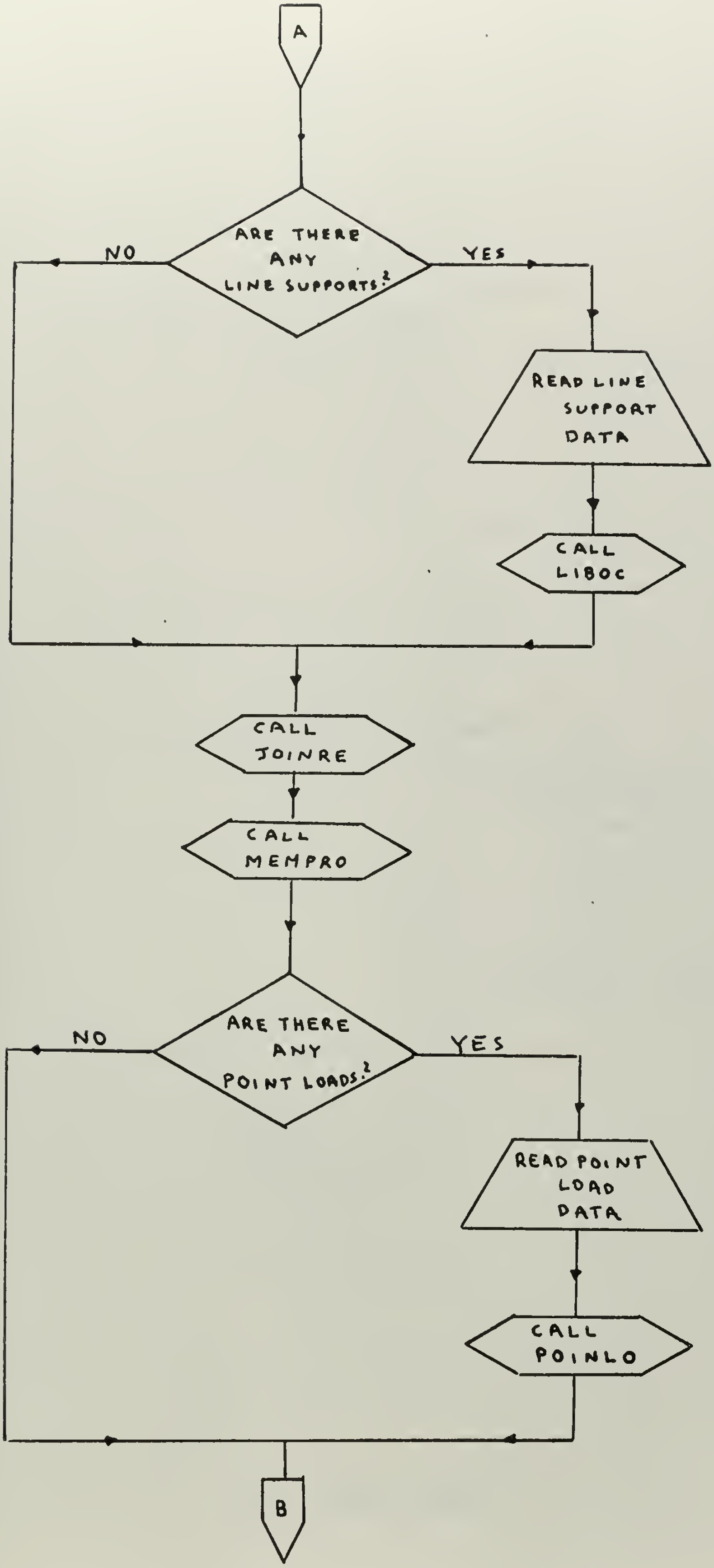

A
ARE THERE ANY LINE SUPPORTS?
NO
YES
READ LINE SUPPORT DATA
CALL LIBOC
CALL JOINRE
CALL MEMPRO
ARE THERE ANY POINT LOADS?
NO
YES
READ POINT LOAD DATA
CALL POINLO
B

Fig. (A5.1) cont.

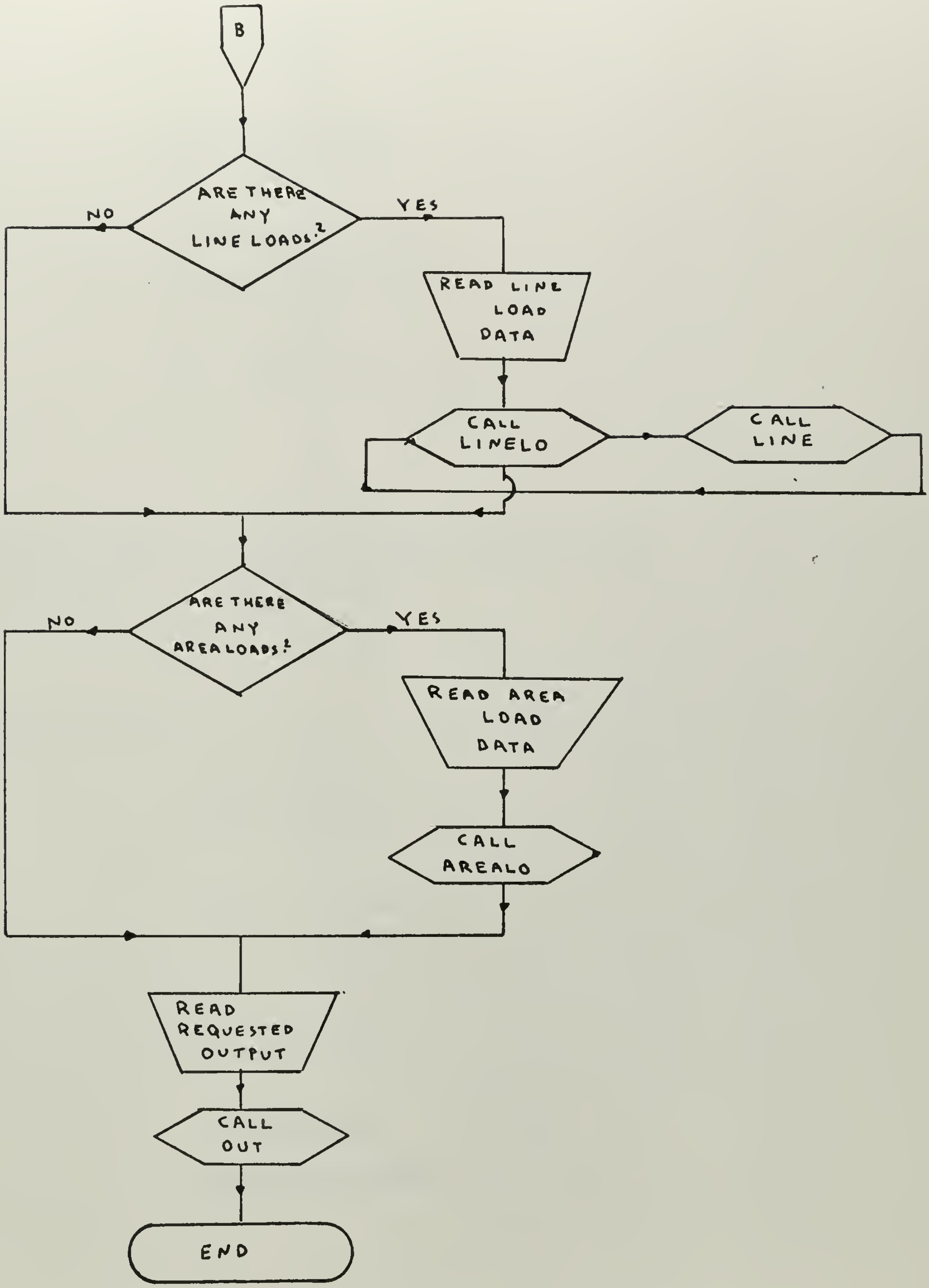

B
ARE THERE ANY LINE LOADS?
NO
YES
READ LINE LOAD DATA
CALL LINELO
CALL LINE
ARE THERE ANY AREA LOADS?
NO
YES
READ AREA LOAD DATA
CALL AREALO
READ REQUESTED OUTPUT
CALL OUT
END

Fig. (A5.1) cont.

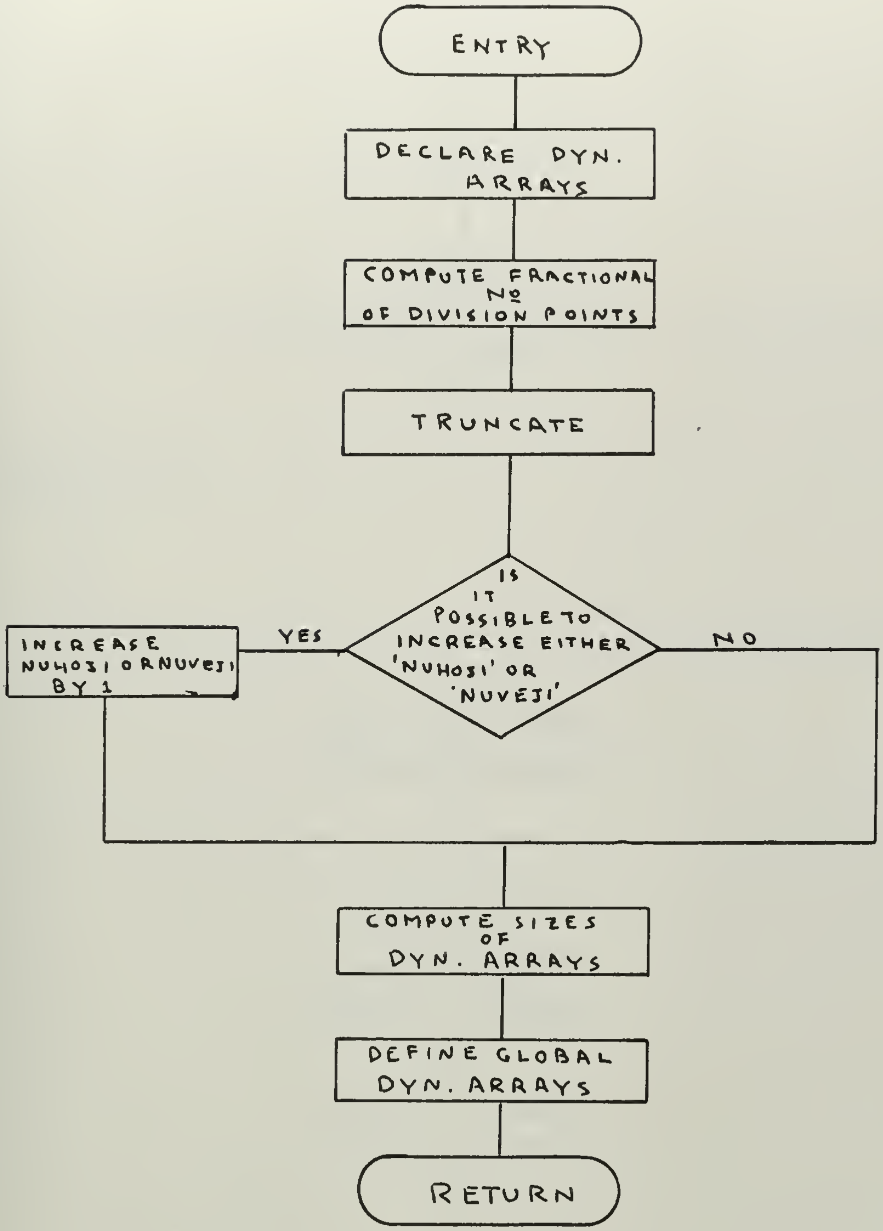

ENTRY
DECLARE DYN. ARRAYS
COMPUTE FRACTIONAL No OF DIVISION POINTS
TRUNCATE
IS IT POSSIBLE TO INCREASE EITHER 'NUHOSI' OR 'NUVEJI'
YES
NO
INCREASE NUHOSI OR NUVEJI BY 1
COMPUTE SIZES OF DYN. ARRAYS
DEFINE GLOBAL DYN. ARRAYS
RETURN

Fig. (A5.2)

SUBROUTINE DIPONH

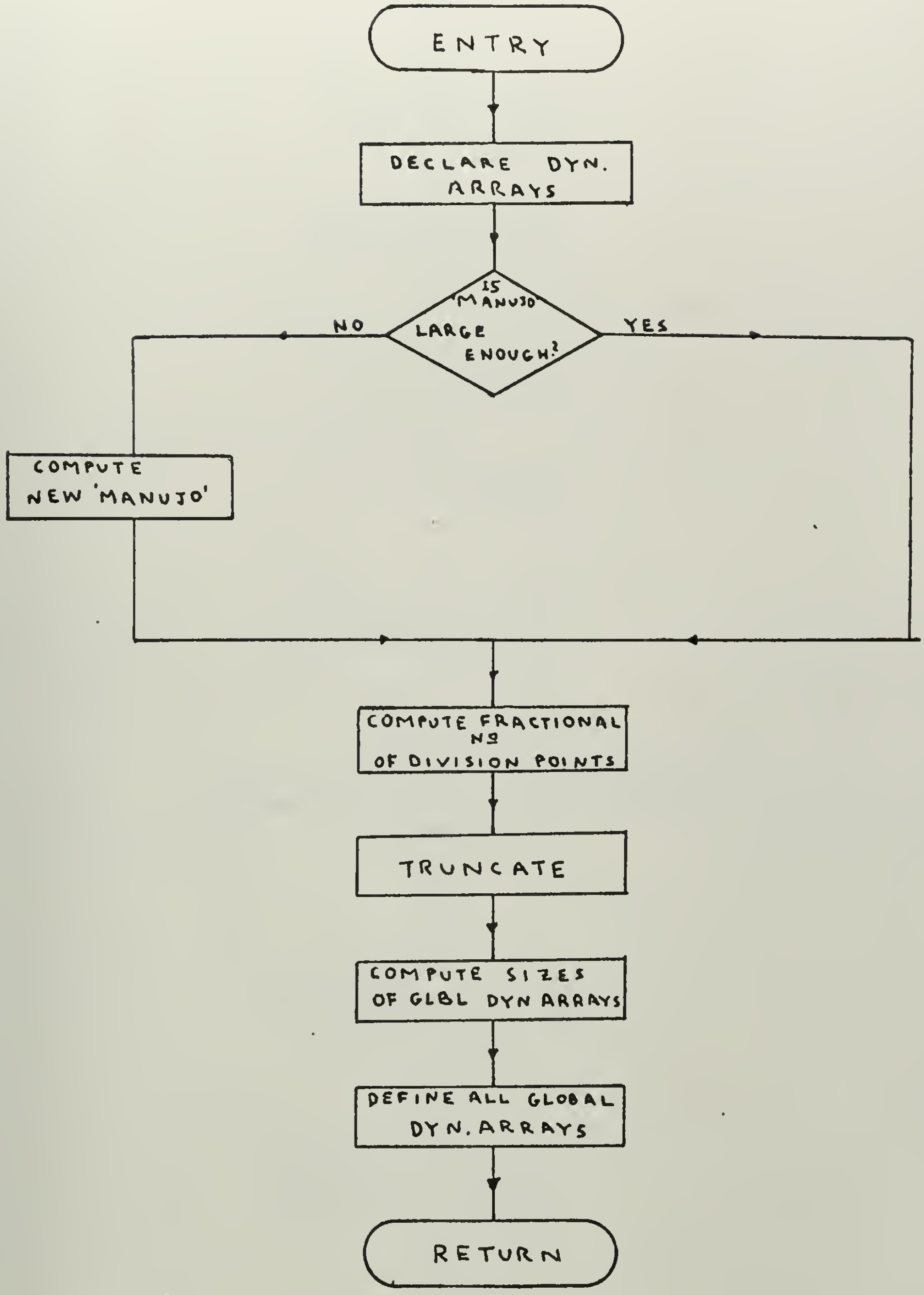

ENTRY
DECLARE DYN. ARRAYS
IS 'MANUJO' LARGE ENOUGH?
NO
YES
COMPUTE NEW 'MANUJO'
COMPUTE FRACTIONAL Nº OF DIVISION POINTS
TRUNCATE
COMPUTE SIZES OF GLBL DYN ARRAYS
DEFINE ALL GLOBAL DYN. ARRAYS
RETURN

Fig. (A5.3)

SUBROUTINE DIPOHO

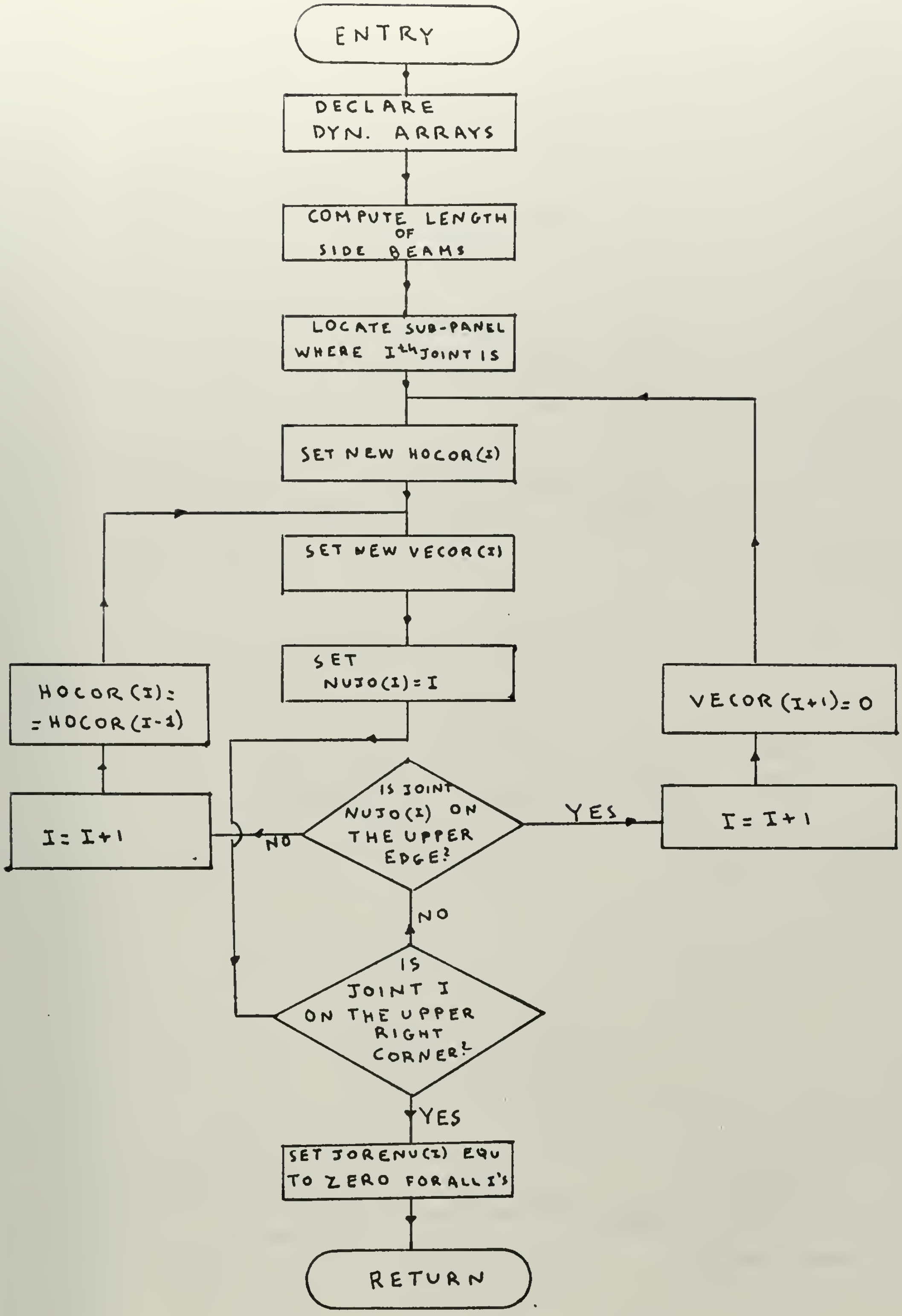

ENTRY
DECLARE DYN. ARRAYS
COMPUTE LENGTH OF SIDE BEAMS
LOCATE SUB-PANEL WHERE Ith JOINT IS
SET NEW HOCOR(I)
SET NEW VECOR(I)
SET NUJO(I)=I
HOCOR(I):=HOCOR(I-1)
I:=I+1
IS JOINT NUJO(I) ON THE UPPER EDGE?
NO
YES
I=I+1
VECOR(I+1)=0
NO
IS JOINT I ON THE UPPER RIGHT CORNER?
YES
SET JORENU(I) EQU TO ZERO FOR ALL I's
RETURN

Fig. (A5.5)

SUBROUTINES GRID AND MUGRID

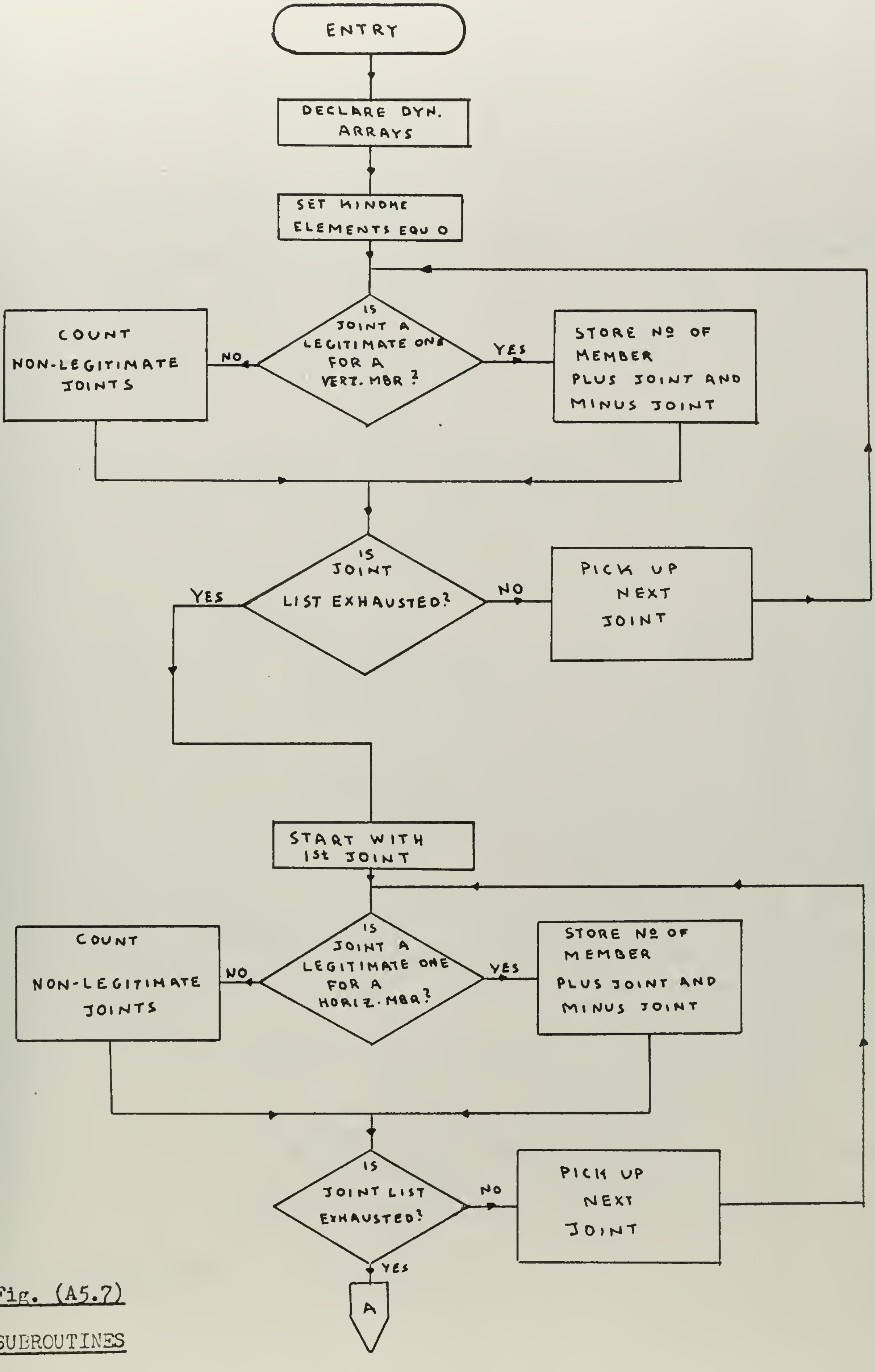

ENTRY
DECLARE DYN. ARRAYS
SET KINOME ELEMENTS EQU 0
IS JOINT A LEGITIMATE ONE FOR A VERT. MBR ?
NO
YES
COUNT NON-LEGITIMATE JOINTS
STORE No OF MEMBER PLUS JOINT AND MINUS JOINT
IS JOINT LIST EXHAUSTED?
YES
NO
PICK UP NEXT JOINT
START WITH 1st JOINT
IS JOINT A LEGITIMATE ONE FOR A HORIZ. MBR?
NO
YES
COUNT NON-LEGITIMATE JOINTS
STORE No OF MEMBER PLUS JOINT AND MINUS JOINT
IS JOINT LIST EXHAUSTED?
NO
PICK UP NEXT JOINT
YES
A

Fig. (A5.7)

SUBROUTINES

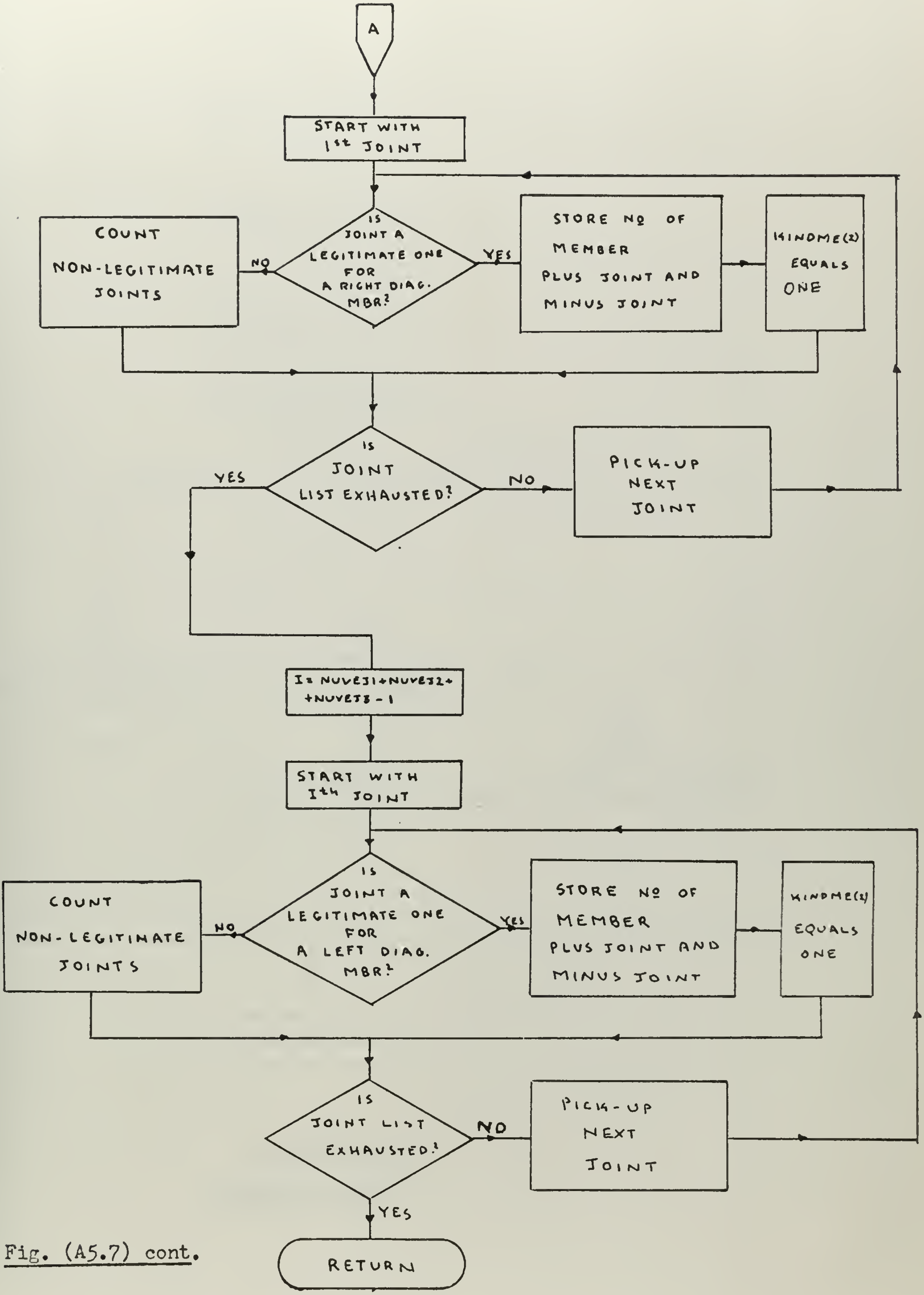

A
START WITH 1st JOINT
IS JOINT A LEGITIMATE ONE FOR A RIGHT DIAG. MBR?
NO
COUNT NON-LEGITIMATE JOINTS
YES
STORE No OF MEMBER PLUS JOINT AND MINUS JOINT
KINDME(2) EQUALS ONE
IS JOINT LIST EXHAUSTED?
NO
PICK-UP NEXT JOINT
YES
I = NUVEJ1+NUVEJ2+ +NUVEJ3 - 1
START WITH Ith JOINT
IS JOINT A LEGITIMATE ONE FOR A LEFT DIAG. MBR?
NO
COUNT NON-LEGITIMATE JOINTS
YES
STORE No OF MEMBER PLUS JOINT AND MINUS JOINT
KINDME(2) EQUALS ONE
IS JOINT LIST EXHAUSTED?
NO
PICK-UP NEXT JOINT
YES
RETURN

Fig. (A5.7) cont.

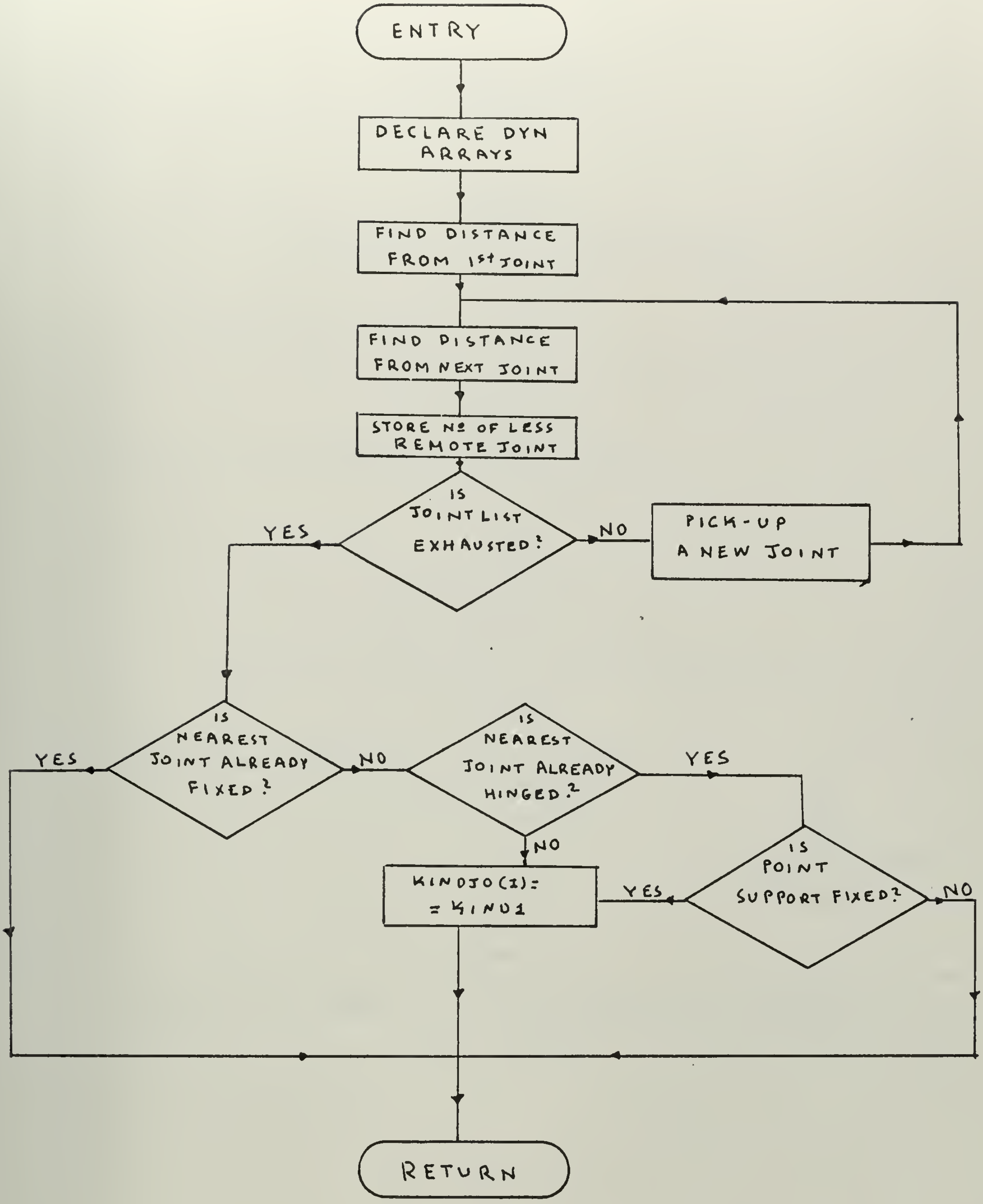

ENTRY
DECLARE DYN ARRAYS
FIND DISTANCE FROM 1st JOINT
FIND DISTANCE FROM NEXT JOINT
STORE No OF LESS REMOTE JOINT
IS JOINT LIST EXHAUSTED?
YES
NO
PICK-UP A NEW JOINT
IS NEAREST JOINT ALREADY FIXED?
YES
NO
IS NEAREST JOINT ALREADY HINGED?
YES
NO
KINDJO(I)= =KIND1
YES
IS POINT SUPPORT FIXED?
NO
RETURN

Fig. (A5.8)

SUBROUTINE POBOC

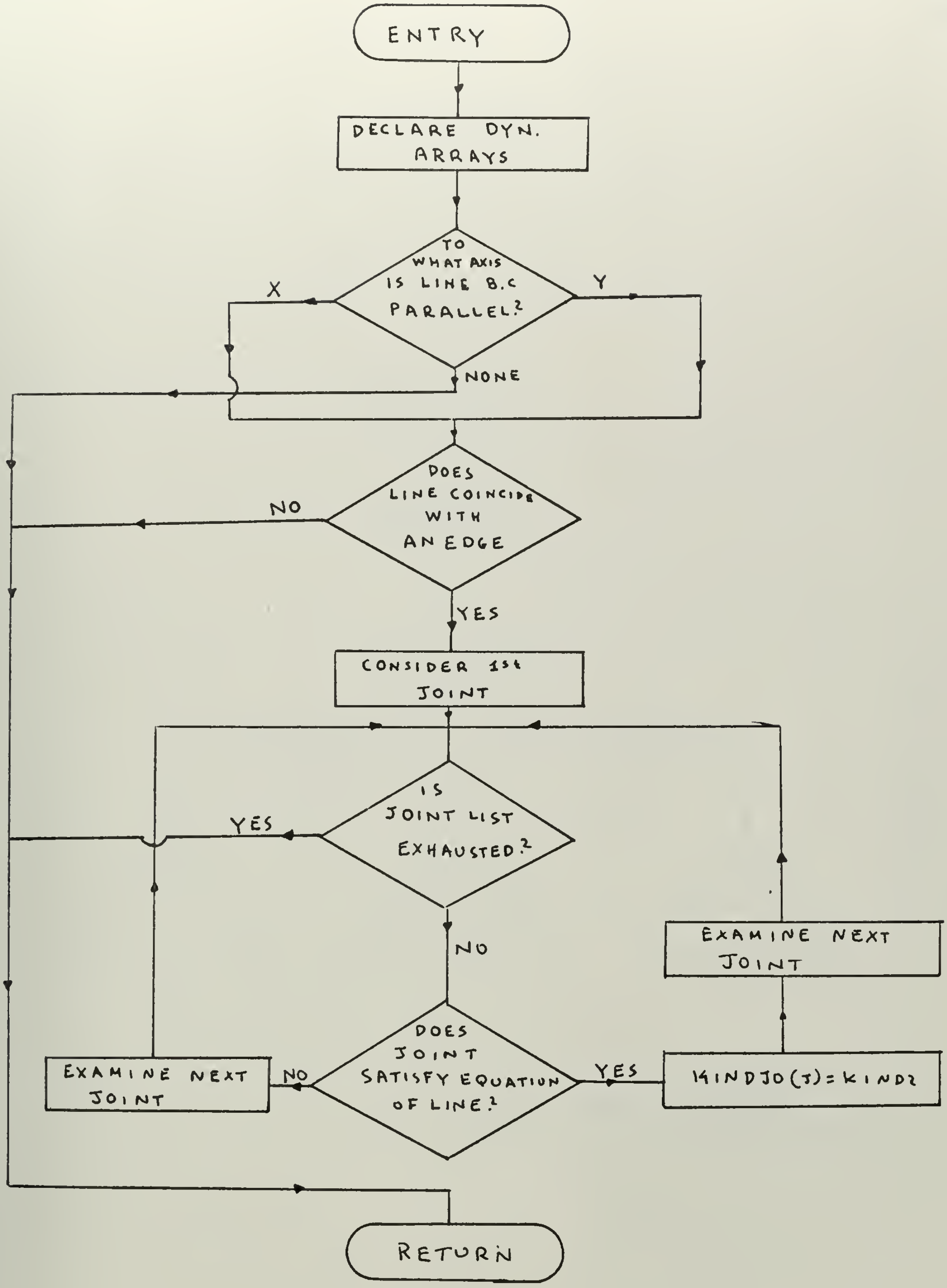


Fig. (A5.9)

SUBROUTINE LIBOC

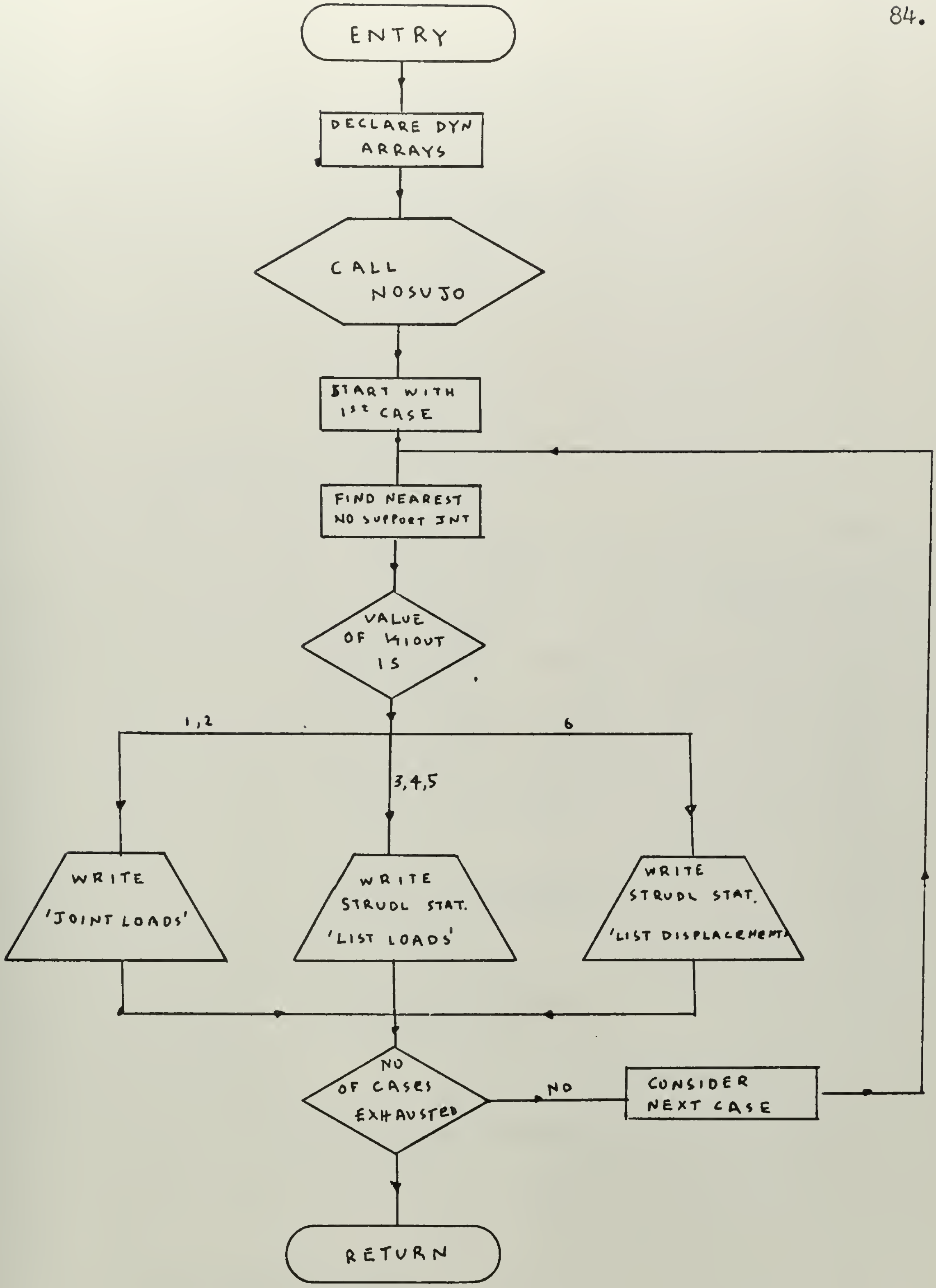


Fig. (A5.10)

SUBROUTINE POINLO

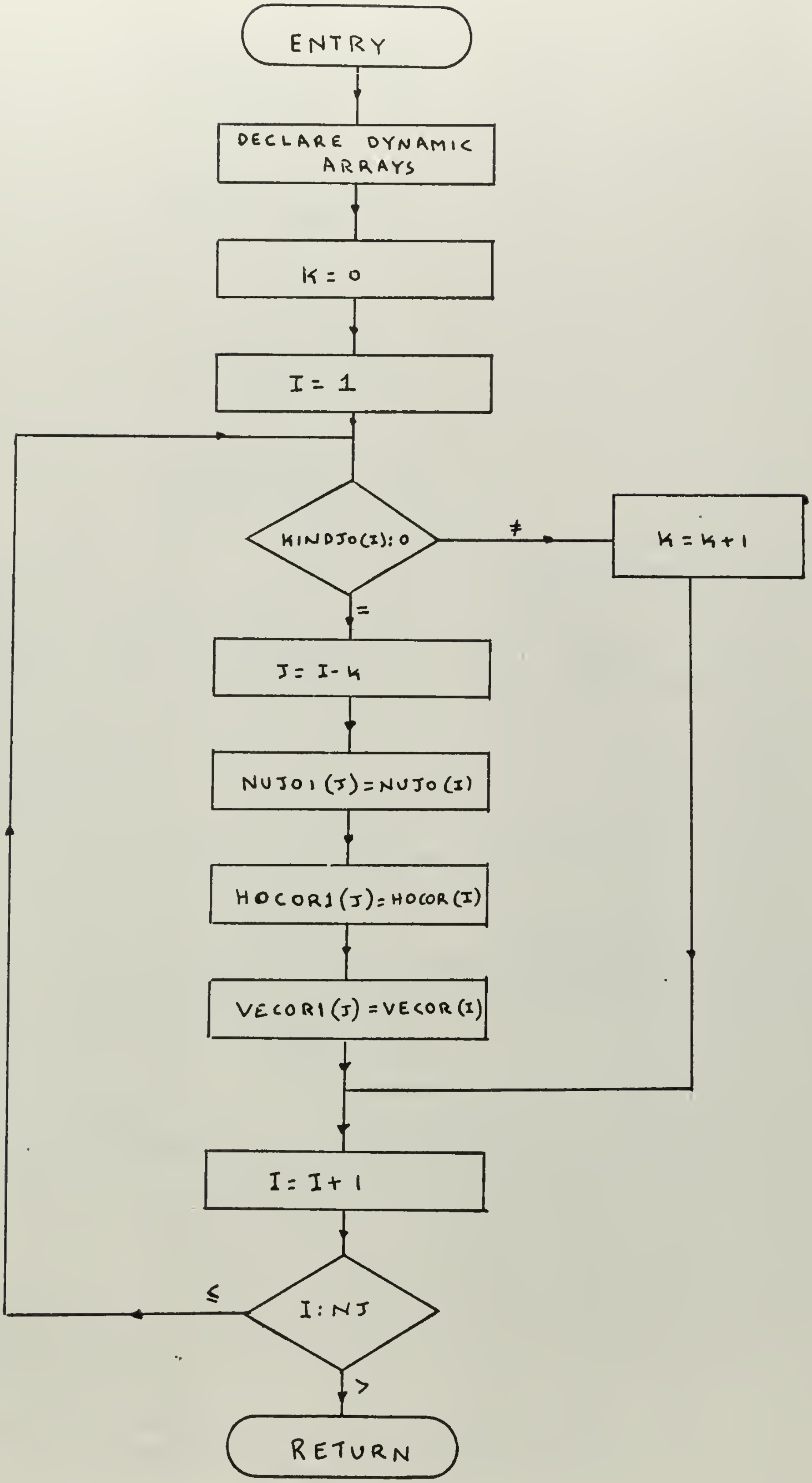


Fig. (A5.11)
SUBROUTINE NOSUJO

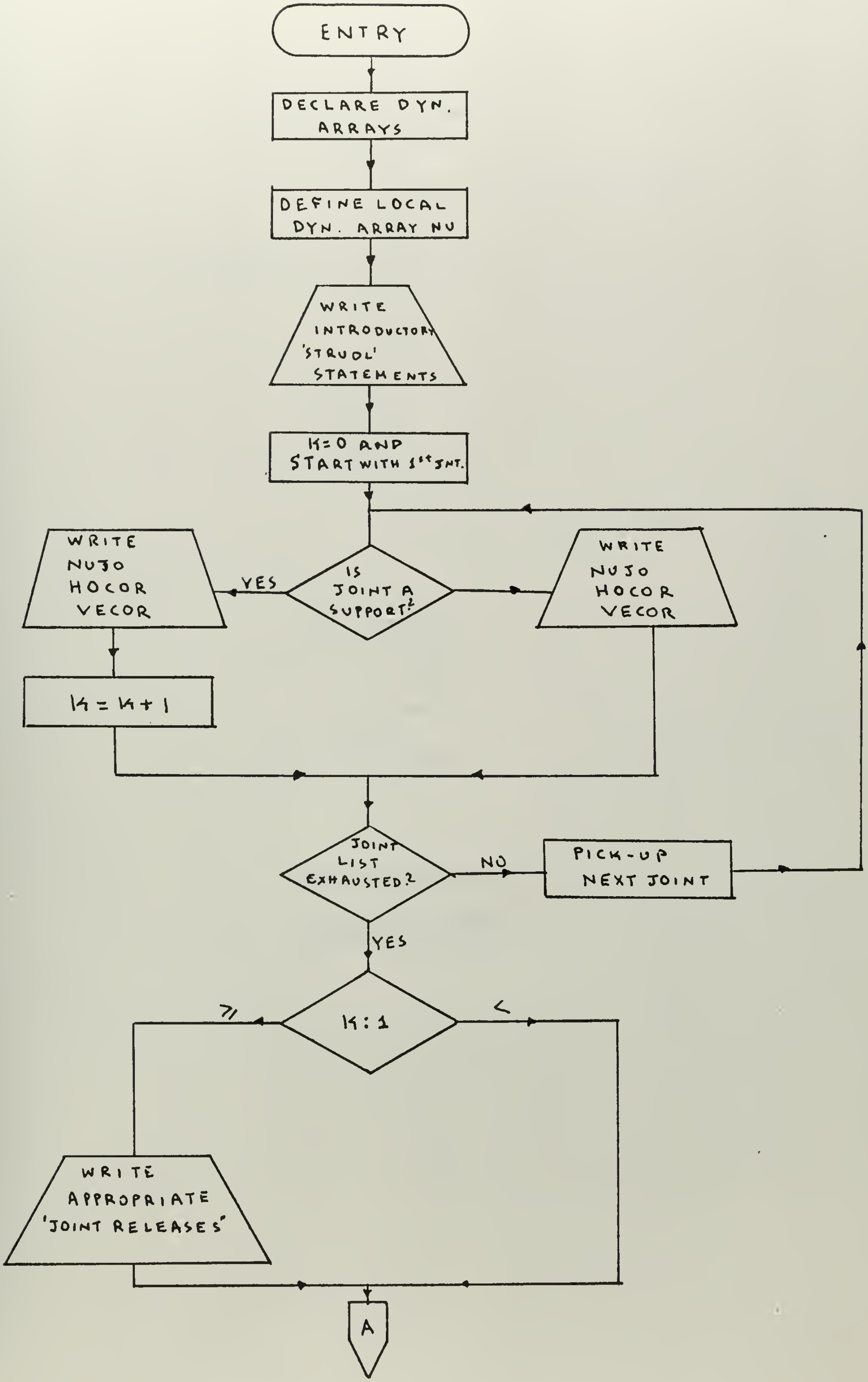


Fig. (A5.12)

SUBROUTINE JOINRE

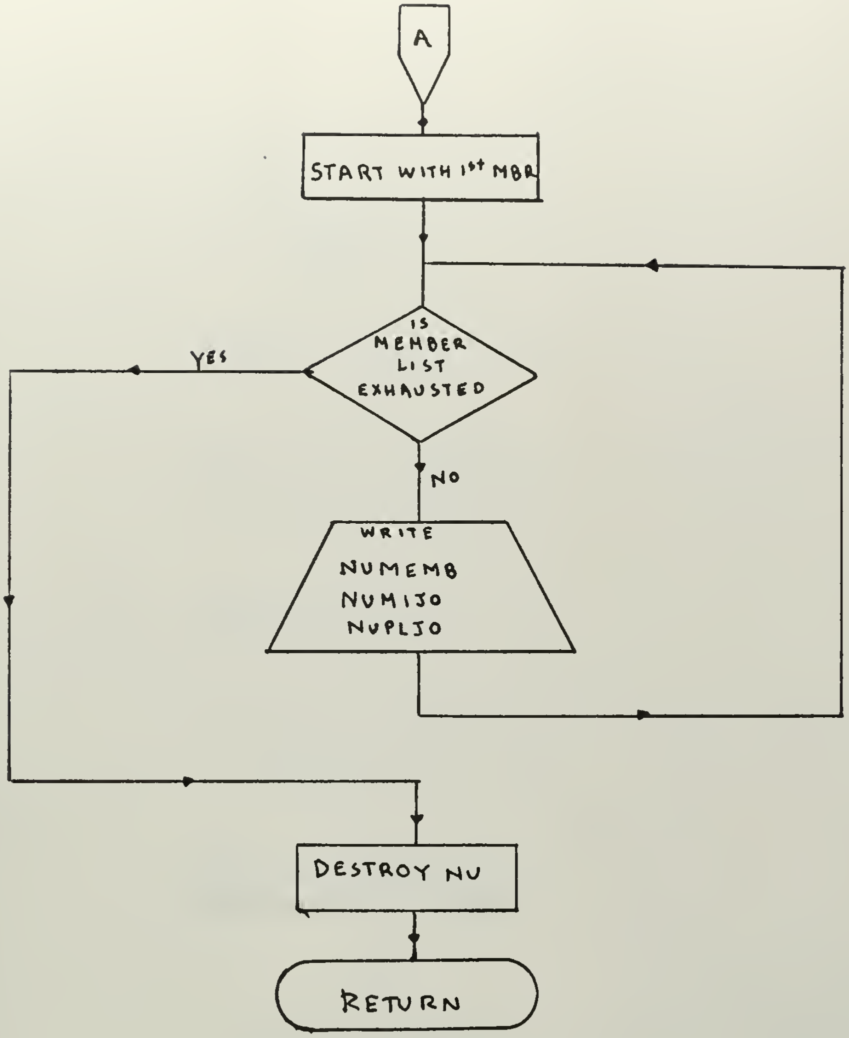


Fig. (A5.12) cont.

SUBROUTINE JOINRE

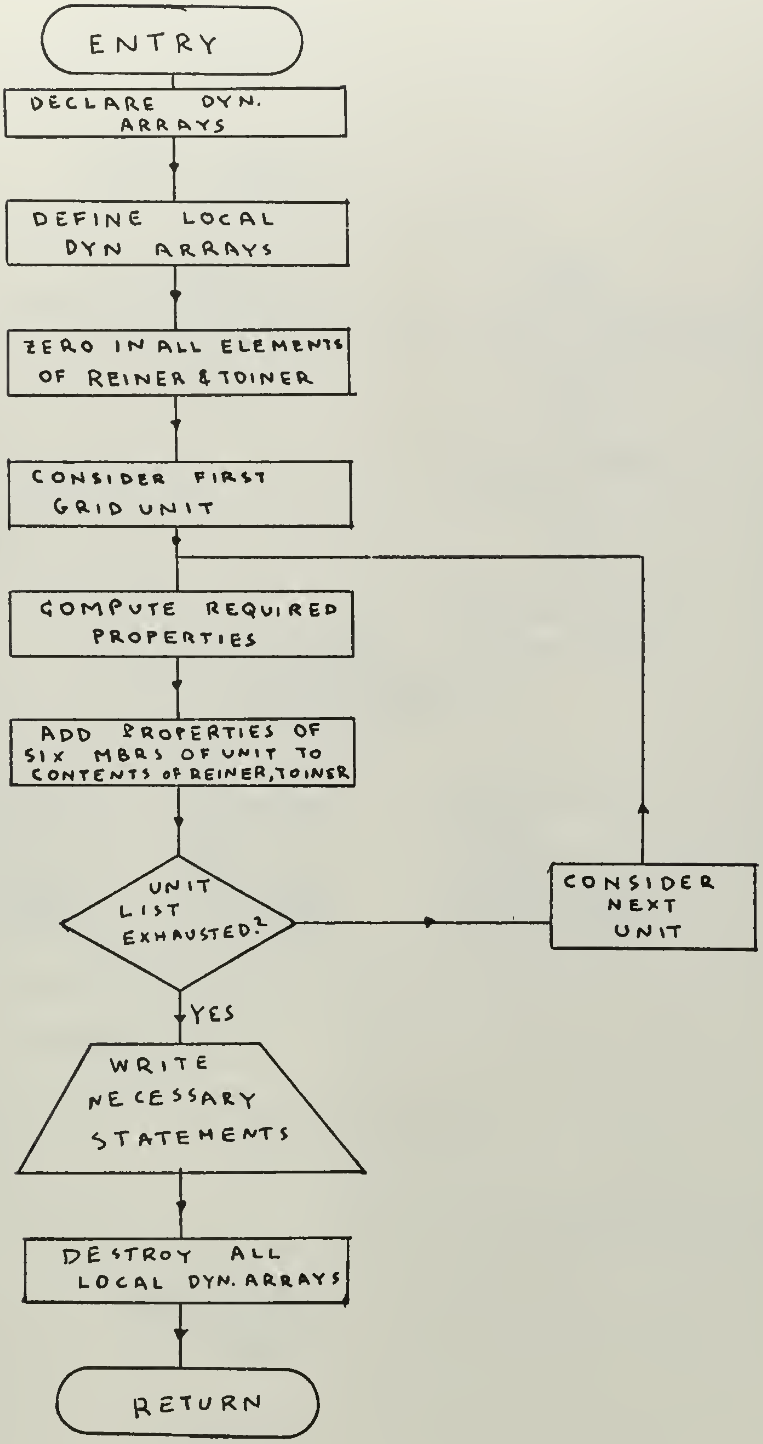

ENTRY
DECLARE DYN. ARRAYS
DEFINE LOCAL DYN ARRAYS
ZERO IN ALL ELEMENTS OF REINER & TOINER
CONSIDER FIRST GRID UNIT
COMPUTE REQUIRED PROPERTIES
ADD PROPERTIES OF SIX MBRS OF UNIT TO CONTENTS OF REINER, TOINER
UNIT LIST EXHAUSTED?
CONSIDER NEXT UNIT
YES
WRITE NECESSARY STATEMENTS
DESTROY ALL LOCAL DYN. ARRAYS
RETURN

Fig. (A5.13)

SUBROUTINE MEMPRO

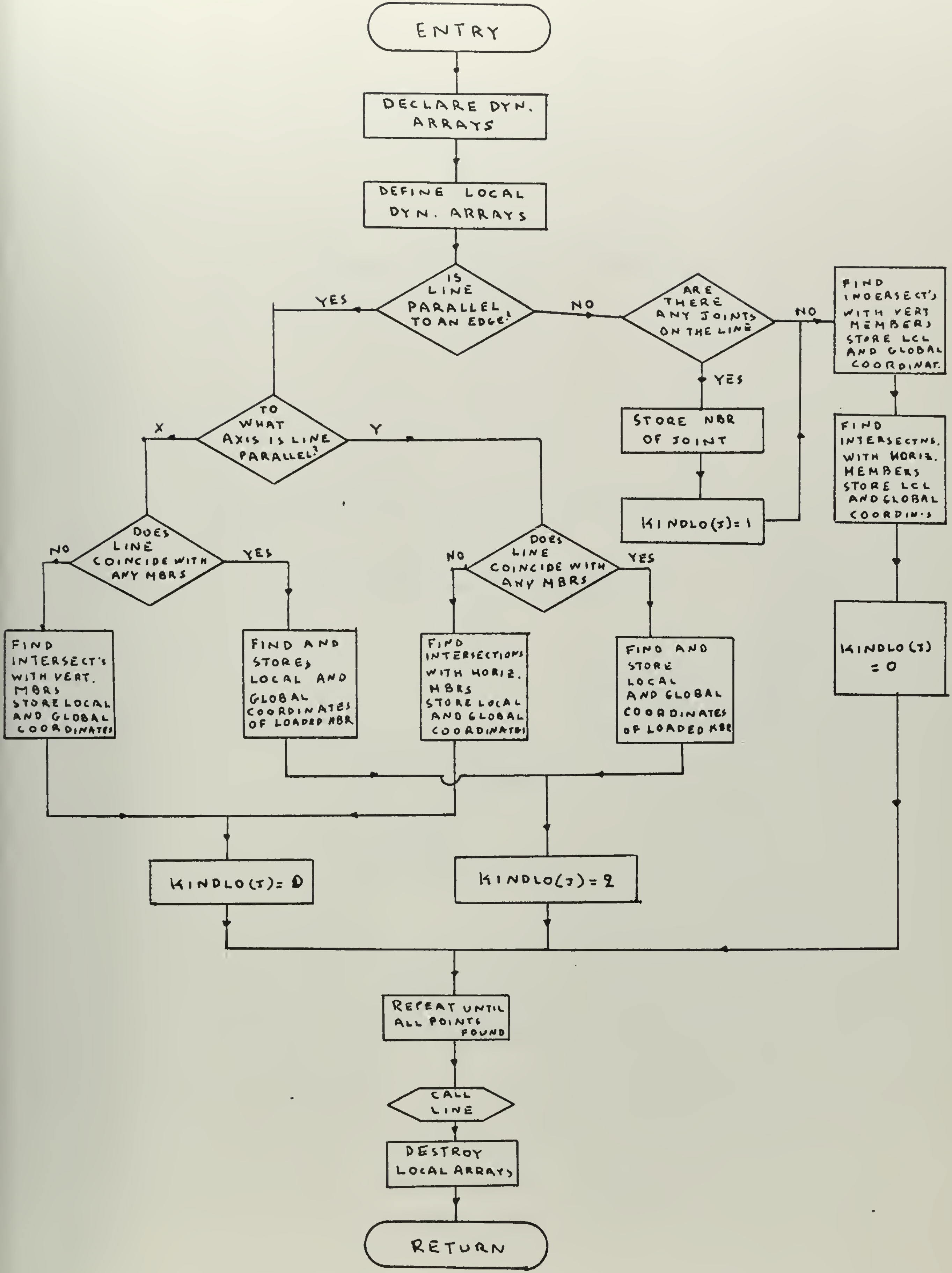

ENTRY
DECLARE DYN. ARRAYS
DEFINE LOCAL DYN. ARRAYS
IS LINE PARALLEL TO AN EDGE?
YES
NO
ARE THERE ANY JOINTS ON THE LINE
NO
FIND INTERSECT'S WITH VERT MEMBERS STORE LCL AND GLOBAL COORDINAT.
YES
STORE NBR OF JOINT
KINDLO(J)=1
FIND INTERSECTNS. WITH HORIZ. MEMBERS STORE LCL AND GLOBAL COORDIN.S
KINDLO(J) = 0
TO WHAT AXIS IS LINE PARALLEL?
X
Y
DOES LINE COINCIDE WITH ANY MBRS
NO
YES
DOES LINE COINCIDE WITH ANY MBRS
NO
YES
FIND INTERSECT'S WITH VERT. MBRS STORE LOCAL AND GLOBAL COORDINATES
FIND AND STORE LOCAL AND GLOBAL COORDINATES OF LOADED MBR
FIND INTERSECTIONS WITH HORIZ. MBRS STORE LOCAL AND GLOBAL COORDINATES
FIND AND STORE LOCAL AND GLOBAL COORDINATES OF LOADED MBR
KINDLO(J)= 0
KINDLO(J)= 2
REPEAT UNTIL ALL POINTS FOUND
CALL LINE
DESTROY LOCAL ARRAYS
RETURN

Fig. (A5.14)

SUBRCUTINE LINELO

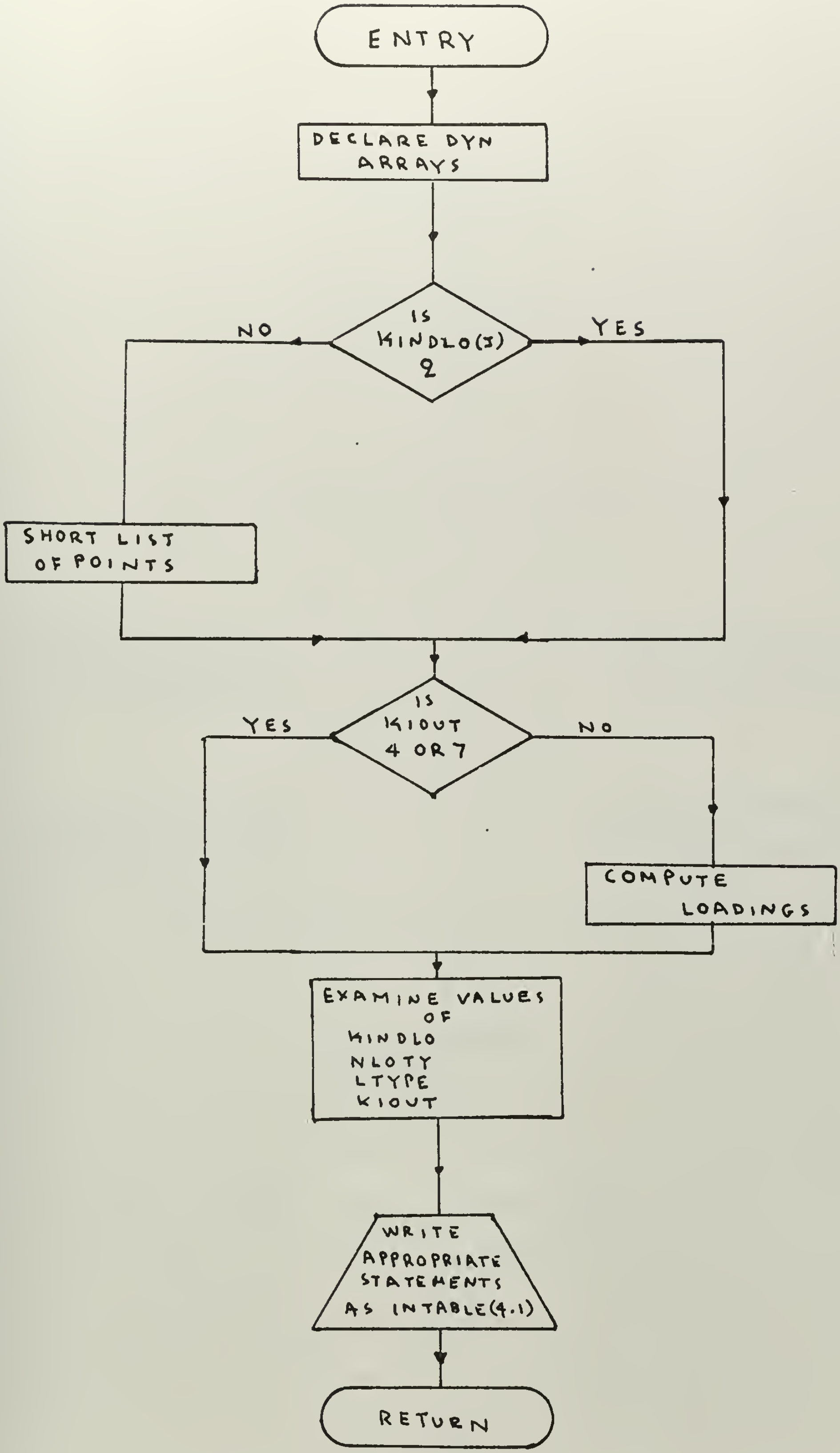


Fig. (A5.15)

SUBROUTINE LINE

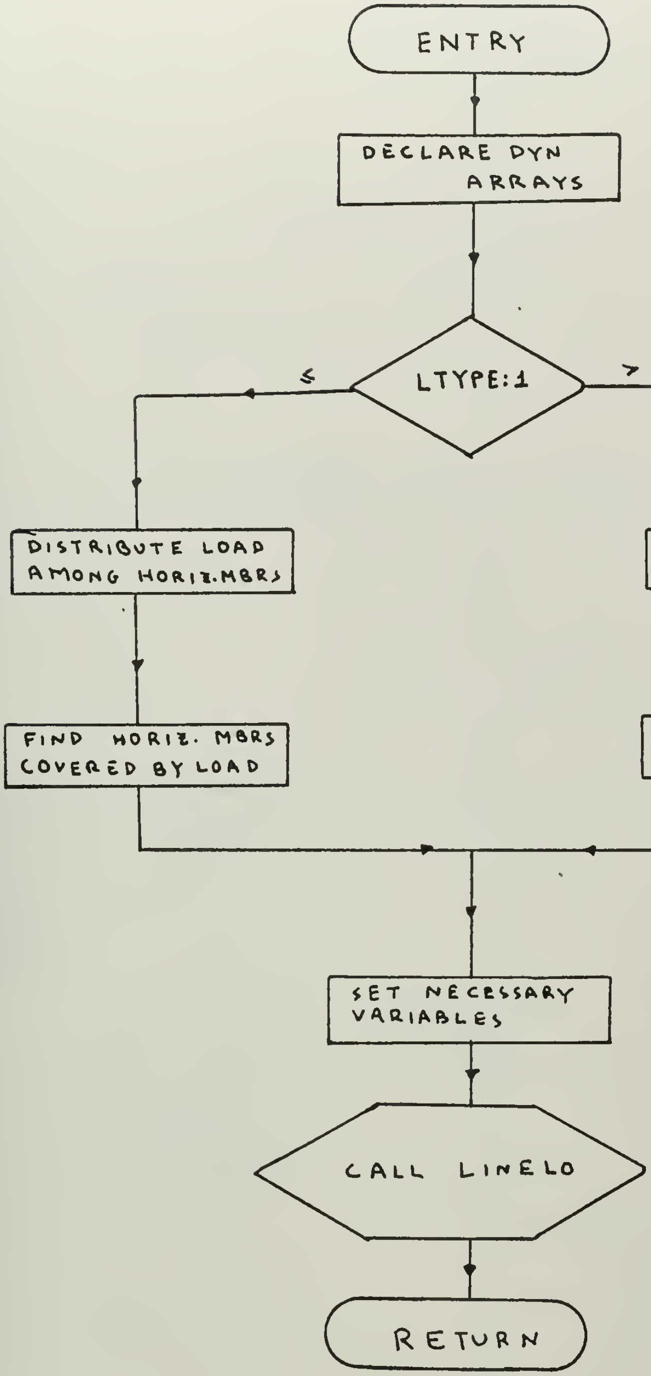

ENTRY
DECLARE DYN ARRAYS
LTYPE:1
≤
>
DISTRIBUTE LOAD AMONG HORIZ.MBRS
DISTRIBUTE LOAD AMONG VERT. MBRS
FIND HORIZ. MBRS COVERED BY LOAD
FIND VERT. MBRS COVERED BY LOAD
SET NECESSARY VARIABLES
CALL LINELO
RETURN

Fig. (A5.16)

SUBROUTINE AREALO

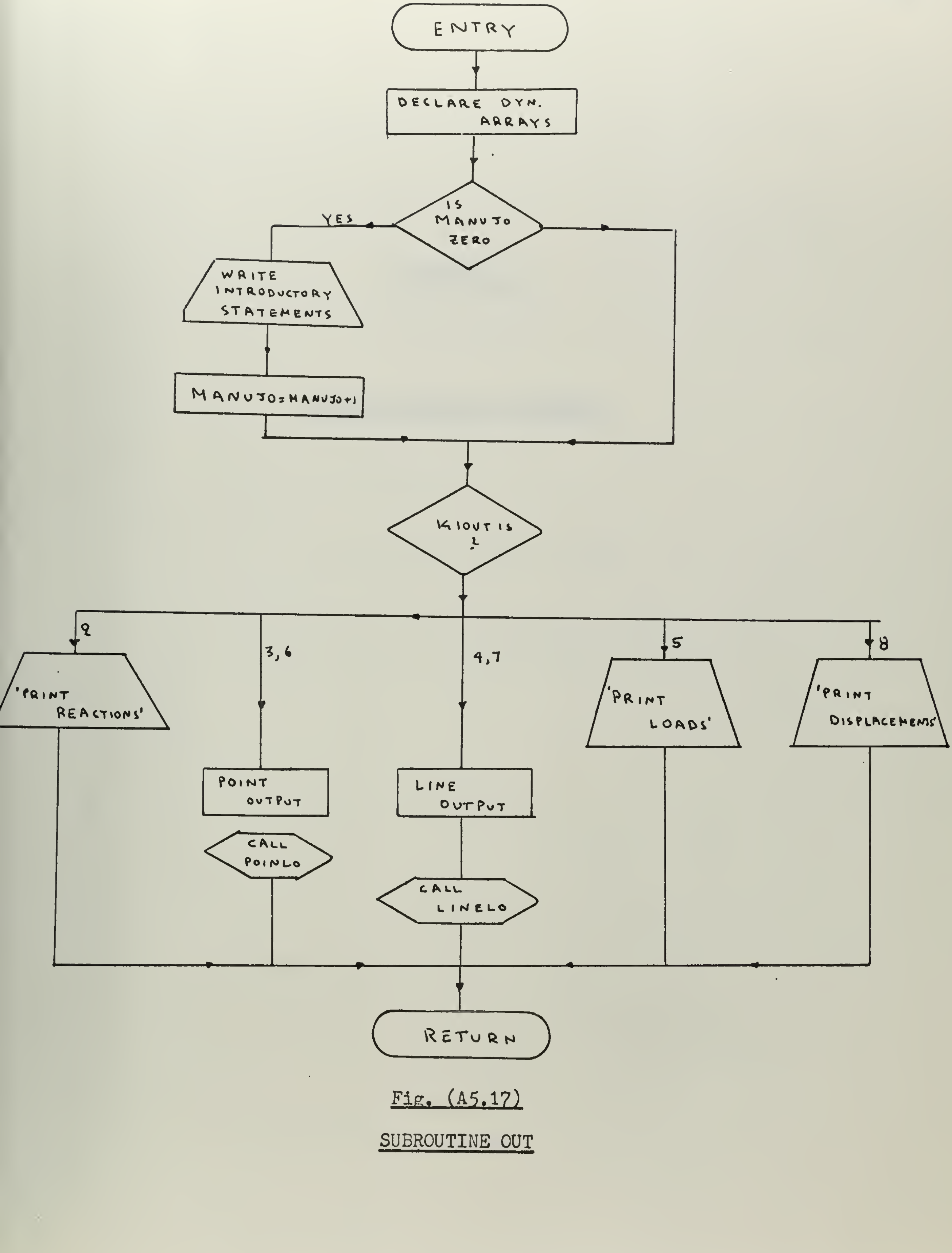


Fig. (A5.17)

SUBROUTINE OUT

APPENDIX B

ICETRAN LISTINGS OF PROGRAMS

```
/JOB           KOLLINIATIS
/RESTORE OLD 69
/LIBRARY ICESLIB
/DUMP
**026
      DYNAMIC ARRAY HOCOR,VECOR,NUJO,JORENU,KINDJO,NUPLJO,NUMIJO,NUMEMB
      DYNAMIC ARRAY KINDME,NUJO1,HOCOR1,VECOR1
      REAL LDMSTA,LDMEND
      COMMON HOLEN,VELEN,THICK,MANUJO,YOUNG,PRATIO,HOLEN1,HOLEN2,VELEN1
     1,VELEN2,EX,WHY,KIND1,EX2,WHY2,POLOM1,KILILO,LTYPE,NUHOJ1,NUVEJ1,
     2NJ,HOCOR(P),VECOR(P),NUJO(P),JORENU(P),HOLEN3,VELEN3,NUHOJ2,NUHOJ3
     3,NUVEJ2,NUVEJ3,NUPLJO(P),NUMIJO(P),NUMEMB(P),KINDME(P),NUME,NUVEMB
     4,NUHOMB,NURIDI,NULEDI,NUPOLO,NUJO1(P),HOCOR1(P),VECOR1(P),NSIZE,
     5NULDS,ENDLD,LDMSTA,LDMEND,KINDJO(P),KIOUT,NLOTY
      EQUIVALENCE (EX,EX1,EXLOAD,XA,OUTXA),(WHY,wHY1,WYLOAD,YA,OUTYA),
     1(KIND1,KIND2),(EX2,XB,OUTXB),(WHY2,YB,OUTYB),(STALD,POLOM1)
    2 FORMAT (5F10.3)
    3 FORMAT (I6)
    4 FORMAT (I4,4F10.3)
    5 FORMAT (2I4)
    6 FORMAT (I4,4F10.3)
    7 FORMAT (3I4)
    8 FORMAT (I5,3F10.3,I5,3F10.3)
    9 FORMAT (I5,4F10.3)
P     CALL INIT(QQDUB)
      READ (5,2) HOLEN,VELEN,THICK,YOUNG,PRATIO
      WRITE(6,2) HOLEN,VELEN,THICK,YOUNG,PRATIO
      READ (5,3) MANUJO
      WRITE(6,3) MANUJO
      READ (5,4) NHOL,HOLEN1,VELEN1,HOLEN2,VELEN2
      WRITE(6,4) NHOL,HOLEN1,VELEN1,HOLEN2,VELEN2
      GO TO (50,60),NHOL
   50 CONTINUE
C     THERE IS NO HOLE
      HOLEN1=HOLEN
      VELEN1=VELEN
      CALL DIPONH
   10 CONTINUE
      CALL GRID
   11 CONTINUE
      CALL MEINNH
   12 CONTINUE
      GO TO 70
   60 CONTINUE
C     THERE IS A HOLE
      CALL DIPOHO
   13 CONTINUE
      CALL MUGRID
   14 CONTINUE
      CALL MEINHO
   70 CONTINUE
      READ (5,5) NUPSU,NULSU
      WRITE(6,5) NUPSU,NULSU
      IF (NUPSU) 85,85,80
   80 CONTINUE
      DO 85 I=1,NUPSU
      READ  (5,6) KIND1,EX,WHY
      WRITE (6,6) KIND1,EX,WHY
      NUPSUP=1
```

```
      CALL POBOC
   85 CONTINUE
      IF (NULSU) 15,15,90
   90 CONTINUE
      DO 15 I=1,NULSU
      READ  (5,6) KIND2,EX1,WHY1,EX2,WHY2
      WRITE (6,6) KIND2,EX1,WHY1,EX2,WHY2
      NULSUP=1
      CALL LIBOC
   15 CONTINUE
      CALL JOINRE
   16 CONTINUE
      CALL MEMPRO
  100 CONTINUE
      READ (5,7) NUPOL,NULIL,NUARL
      WRITE(6,7) NUPOL,NULIL,NUARL
      IF (NUPOL) 110,110,105
  105 CONTINUE
      DO 110 I=1,NUPOL
      KIOUT=1
      NUPOLO=1
      READ  (5,9) NLOTY,EXLOAD,WYLOAD,POLOM1
      WRITE (6,9) NLOTY,EXLOAD,WYLOAD,POLOM1
      CALL POINLO
  110 CONTINUE
      IF (NULIL) 120,120,115
  115 CONTINUE
      DO 120 I=1,NULIL
      KIOUT=1
      READ (5,8) NLOTY,XA,YA,STALD,LTYPE,XB,YB,ENDLD
      WRITE(6,8) NLOTY,XA,YA,STALD,LTYPE,XB,YB,ENDLD
      CALL LINELO
  120 CONTINUE
      IF (NUARL) 130,130,125
  125 CONTINUE
      DO 130 I=1,NUARL
      KIOUT=1
      READ (5,8) NLOTY,XA,YA,LDMSTA,LTYPE,XB,YB,LDMEND
      WRITE(6,8) NLOTY,XA,YA,LDMSTA,LTYPE,XB,YB,LDMEND
      CALL AREALO
  130 CONTINUE
      READ  (5,5) NUOUT
      WRITE (6,5) NUOUT
      DO 140 I=1,NUOUT
      READ (5,9) KIOUT,OUTXA,OUTYA,OUTXB,OUTYB
      WRITE(6,9) KIOUT,OUTXA,OUTYA,OUTXB,OUTYB
      CALL OUT
  140 CONTINUE
  141 FORMAT (T8,8H FINISH )
      WRITE (7,141)
      DESTROY HOCOR
      DESTROY VECOR
      DESTROY NUJO
      DESTROY NUMIJO
      DESTROY NUPLJO
      DESTROY NUMEMB
      DESTROY KINDME
      DESTROY KINDJO
      DESTROY JORENU
```

```
DESTROY NUJO1
DESTROY HOCOR1
DESTROY VECOR1
END
```

[illegible]

```
**026
      SUBROUTINE DIPUNM
      DYNAMIC ARRAY HOCOR,VECOR,NUJO,JORENU,KINDJO,NUPLJO,NUMIJO,NUMEMB
      DYNAMIC ARRAY KINDME,NUJO1,HOCOR1,VECOR1
      COMMON HOLEN,VELEN,THICK,MANUJO,YOUNG,PRATIO,HOLEN1,HOLEN2,VELEN1
     1,VELEN2,EX,WHY,KIND1,EX2,WHY2,POLOM1,KILILO,LTYPE,NUHOJ1,NUVEJ1,
     2NJ,HOCOR(P),VECOR(P),NUJO(P),JORENU(P),HOLEN3,VELEN3,NUHOJ2,NUHOJ3
     3,NUVEJ2,NUVEJ3,NUPLJO(P),NUMIJO(P),NUMEMB(P),KINDME(P),NUME,NUVEMB
     4,NUHOMB,NURIDI,NULEDI,NUPOLO,NUJO1(P),HOCOR1(P),VECOR1(P),NSIZE,
     5NULDS,ENDLD,LDMSTA,LDMEND,KINDJO(P),KIOUT,NLOTY
      IF (PRATIO-1./3.) 2,2,1
    1 WRITE (6,4)
      CALL EXIT
    2 CONTINUE
    4 FORMAT (41H POISSON'S RATIO LARGE,ANALOGY NOT VALID )
      RATIO=VELEN1/HOLEN1
      ANUJO=MANUJO
      AUVEJ1=SQRT)ANUJO*RATIO*
      AUHOJ1=AUVEJ1/RATIO
      NUVEJ1=AUVEJ1
      NUHOJ1=AUHOJ1
      INCVER=(NUVEJ1+1)*NUHOJ1
      INCHOR=NUVEJ1*(NUHOJ1+1)
      IF (MANUJO-INCVER) 10,5,10
    5 CONTINUE
C     MAX NUMBER OF JOINTS ACHEIVED
      NUVEJ1=NUVEJ1+1
      GO TO 50
   10 CONTINUE
C     EXAMINE POSSIBILITY OF INCREASING HORIZ.JOINTS
      IF (MANUJO-INCHOR) 25,20,30
   20 CONTINUE
C     MAX NUMBER OF JOINTS ACHEIVED
      NUHOJ1=NUHOJ1+1
      GO TO 50
   25 CONTINUE
C     IMPOSSIBLE TO INCREASE EITHER THE HORIZ OR VERT JOINTS
      GO TO 50
   30 CONTINUE
C     EITHER THE HORIZ OR VERT JOINTS CAN BE INCREASED.MAKE LARGER CHANGE
      IF (INCHOR-INCVER)40,35,35
   35 NUHOJ1=NUHOJ1+1
      GO TO 50
   40 NUVEJ1=NUVEJ1+1
   50 CONTINUE
      NUHOMB=(NUHOJ1-1)*NUVEJ1
      NUVEMB=(NUVEJ1-1)*NUHOJ1
      NURIDI=(NUHOJ1-1)*(NUVEJ1-1)
      NULEDI=NURIDI
      NUME=NUHOMB+NUVEMB+NURIDI+NULEDI
      NJ=NUVEJ1*NUHOJ1
      DEFINE KINDME,NUME,FULL,HIGH
      DEFINE NUMEMB,NUME,FULL,HIGH
      DEFINE NUMIJO,NUME,FULL,HIGH
      DEFINE NUPLJO,NUME,FULL,HIGH
```

```
DEFINE VECOR1,NJ,FULL,HIGH
DEFINE HOCOR1,NJ,FULL,HIGH
DEFINE NUJO1 ,NJ,FULL,HIGH
DEFINE KINDJO,NJ,FULL,HIGH
DEFINE JORENU,NJ,FULL,HIGH
DEFINE NUJO,NJ,FULL,HIGH
DEFINE VECOR ,NJ,FULL,HIGH
DEFINE HOCOR ,NJ,FULL,HIGH
MANUJO=0
RETURN
END
```

```
**J26
      SUBROUTINE DIPOHO
      DYNAMIC ARRAY HOCOR,VECOR,NUJO.JOREND.KINDJO,NUPLJO,NUMIJO,NUMEMB
      DYNAMIC ARRAY KINDME,NUJO1,HOCOR1,VECOR1
      COMMON HOLEN,VELEN.THICK,MANUJO,YOUNG,PRATIO,HOLEN1,HOLEN2,VELEN1
     1,VELEN2,EX,WHY.KINDI,EX2,WHY2,POLOM1,KILILO,LTYPE,NUHOJ1,NUVEJ1,
     2NJ,HOCOR(P),VECOR(P),NUJO(P),JOREND(P),HOLEN3,VELEN3,NUHOJ2,NUHOJ3
     3,NUVEJ2,NUVEJ3,NUPLJO(P),NUMIJO(P),NUMEMB(P).KINDME(P),NUME,NUVEMB
     4,NUHOMB,NURIDI,NULEDI,NUPOLO,NUJO1(P),HOCOR1(P),VECOR1(P),NSIZE,
     5NULDS,ENDLD,LDMSTA,LDMEND,KINDJO(P).KIOUT,NLOTY
      DIMENSION D(6),N(6),AN(6),A(6)
      REAL D
C     THIS.SUBROUTINR IS CALLED BY FOGRID.IT GIVES THE NUMBER OF JOINTS
C     FOR EACH PART OF THE PLATE IF THE MAX NUMBER OF JOINTS IS
C     SPECIFIED.THERE IS A HOLE
      IF (PRATIO-1./3.) 2,2,1
    1 WRITE (6,4)
      CALL EXIT
    2 CONTINUE
    4 FORMAT (41H POISSON'S RATIO LARGE,ANALOGY NOT VALID )
      HOLEN3=HOLEN-HOLEN1-HOLEN2
      VELEN3=VELEN-VELEN1-VELEN2
      D(1)=HOLEN1
      D(2)=HOLEN2
      D(3)=HOLEN3
      D(4)=VELEN1
      D(5)=VELEN2
      D(6)=VELEN3
      DO 5 K=1,6
      A(K)=D(K)/D(1)
    5 CONTINUE
      ALPHA=A(4)+A(5)+A(6)+A(2)*A(4)+A(2)*A(6)+A(3)*A(4)+A(3)*A(5)+
     1A(3)*A(6)
      BETA=1.+A(3)+A(4)+A(6)
      ANUJO=MANUJO
      DO 8 K=1,6
      ANUJ=4*ALPHA*((A(1)/A(K))**2)
      IF (ANUJO-ANUJ) 7,8,8
    7 ANUJO=ANUJ
      MANUJO=ANUJO
    8 CONTINUE
      DO 10 K=1,6
      AN)1*=BETA/ALPHA.SQRT)))BETA/ALPHA***2*.ANUJO/ALPHA*
      AN(K)=AN(1)*A(K)
      N(K)=AN(K)
   10 CONTINUE
      N1=N(1)+N(2)+N(3)-2
      N2=N(4)+N(5)+N(6)-2
      NJ=N1*N2-N(2)*N(5)+2*(N(2)+N(5))-4
      NUHOJ1=N(1)
      NUHOJ2=N(2)
      NUHOJ3=N(3)
      NUVEJ1=N(4)
      NUVEJ2=N(5)
      NUVEJ3=N(6)
```

```
NUHOJO=NUHOJ1+NUHOJ2+NUHOJ3-2
NUVEJO=NUVEJ1+NUVEJ2+NUVEJ3-2
NUVEMB=(NUVEJO-1)*NUHOJO-(NUVEJ2-1)*(NUHOJ2-2)
NUHOMB=(NUHOJO-1)*(NUVEJO)-(NUVEJ2-2)*(NUHOJ2-1)
NURIDI=(NUVEJO-1)*(NUHOJO-1)-(NUHOJ2-1)*(NUVEJ2-1)
NULEDI=NURIDI
NUME=NUHOMB+NUVEMB+NURIDI+NULEDI
DEFINE KINDME,NUME,FULL,HIGH
DEFINE NUMEMB,NUME,FULL,HIGH
DEFINE NUMIJO,NUME,FULL,HIGH
DEFINE NUPLJO,NUME,FULL,HIGH
DEFINE VECOR1,NJ,FULL,HIGH
DEFINE HOCOR1,NJ,FULL,HIGH
DEFINE NUJO1 ,NJ,FULL,HIGH
DEFINE KINDJO,NJ,FULL,HIGH
DEFINE JORENU,NJ,FULL,HIGH
DEFINE NUJO,NJ,FULL,HIGH
DEFINE VECOR ,NJ,FULL,HIGH
DEFINE HOCOR ,NJ,FULL,HIGH
MANUJO=0
RETURN
END
```

```
**026
      SUBROUTINE GRID
      DYNAMIC ARRAY HOCOR,VECOR,NUJO,JORENU,KINDJO,NUPLJO,NUMIJO,NUMEMB
      DYNAMIC ARRAY KINDME,NUJO1,HOCOR1,VECOR1
      COMMON HOLEN,VELEN,THICK,MANUJO,YOUNG,PRATIO,HOLEN1,HOLEN2,VELEN1
     1,VELEN2,EX,WHY,KIND1,EX2,WHY2,POLOM1,KILILO,LTYPE,NUHOJ1,NUVEJ1,
     2NJ,HOCOR(P),VECOR(P),NUJO(P),JORENU(P),HOLEN3,VELEN3,NUHOJ2,NUHOJ3
     3,NUVEJ2,NUVEJ3,NUPLJO(P),NUMIJO(P),NUMEMB(P),KINDME(P),NUME,NUVEMB
     4,NUHOMB,NURIDI,NULEDI,NUPOLO,NUJO1(P),HOCOR1(P),VECOR1(P),NSIZE,
     5NULDS,ENDLD,LDMSTA,LDMEND,KINDJO(P),KIOUT,NLOTY
      EQUIVALENCE (NUHOJ1,NUHOJO),(NUVEJ1,NUVEJO)
C     THIS SUBROUTINE GENERATES A GRID,GIVEN A RECT. PLATE WITHOUT A HOLE
      NVESPA=NUVEJO-1
      NHOSPA=NUHOJO-1
      HOSPAN=NHOSPA
      VESPAN=NVESPA
      XSPAC=HOLEN/HOSPAN
      YSPAC=VELEN/VESPAN
      J=0
      NVESPA=NVESPA+1
      NHOSPA=NHOSPA+1
      DO 30 IO=1,NHOSPA
      I=IO-1
      DO 20 JO=1,NVESPA
      J=JO-1
      K=I*NUVEJO+J+1
      AI=I
      IF (I-NUHOJO+1) 12,10,12
   10 HOCOR(K)=HOLEN
      GO TO 13
   12 CONTINUE
      HOCOR(K)=AI*XSPAC
   13 CONTINUE
      AJ=J
      IF (J-NUVEJO+1) 15,14,15
   14 VECOR(K)=VELEN
      GO TO 16
   15 CONTINUE
      VECOR(K)=AJ*YSPAC
   16 CONTINUE
      NUJO(K)=K
   20 CONTINUE
      J=0
   30 CONTINUE
      DO 31 J=1,NJ
      JORENU(J)=0
   31 CONTINUE
      RETURN
      END
```

```
[illegible]
      SUBROUTINE [illegible]
      DYNAMIC ARRAY HOCOR,VECOR,NOJN,ELEMNO,KINDJ2,NOPLJC,NORTJC,ELEMNE
      DYNAMIC ARRAY KINDNE,NOJO1,HOCOR1,VECOR1
      COMMON HOLEN,VELEN,TIMEX,ANGLE,YOUNG,POISON,HOLEN1,HOLEN2,VELEN
     1,VELEN2,EX,EY,AINDI,EX2,EY2,POISN1,KJLIEL,LTYPE,NUHOJ1,NUVEJ1,
     2NJ,HOCOR(P),VECOR(P),NOJN(P),ELEMNO(P),HOLEN3,VELEN3,NUHOJ2,NUHOJ
     3,NUVEJ2,NUVEJ3,NOPLJC(P),NORTJC(P),ELEMNE(P),KINDNE(P),NUME,NUVEJ
     4,NUHOJ,NORIJ1,HOLEL,NOPOLS,NOJO1(P),HOCOR1(P),VECOR1(P),NSIZE,
     5NCLDS,ENDLD,EDOUT,ELMENE,KINDJC(P),NIOUT,NEUTY
C     THIS SUBROUTINE GENERATES A GRID GIVEN A RECT.PLATE WITH A HOLE
      AUHOJ1=NUHOJ1
      AUHOJ2=NUHOJ2
      AUHOJ3=NUHOJ3
      XSPAC1=HOLEN1/(AUHOJ1-1.)
      XSPAC2=HOLEN2/(AUHOJ2-1.)
      XSPAC3=HOLEN3/(AUHOJ3-1.)
      NUHOJ=NUHOJ1+NUHOJ2+NUHOJ3-3
      NUVEJ=NUVEJ1+NUVEJ2+NUVEJ3-3
      AUVEJ1=NUVEJ1
      AUVEJ2=NUVEJ2
      AUVEJ3=NUVEJ3
      YSPAC1=VELEN1/(AUVEJ1-1.)
      YSPAC2=VELEN2/(AUVEJ2-1.)
      YSPAC3=VELEN3/(AUVEJ3-1.)
      J=0
      MAX=NUHOJ+1
      DO 30 INDEX=1,MAX
      I=INDEX-1
      NHOSPA=NUHOJ1-1
      AI=I
      J=0
      IF (I-NHOSPA) 5,5,7
    5 CONTINUE
C     TO THE LEFT OF HOLE OR AT ITS LEFT EDGE
      X=AI*XSPAC1
      NSUCRO=I*(NUVEJ1+NUVEJ2+NUVEJ3-2)
      GO TO 15
    7 CONTINUE
C     AT THE HOLE OR TO THE RIGHT OF IT
      NHOSPA=NUHOJ1+NUHOJ2-1
      IF (I-NHOSPA+1) 9,9,11
    9 CONTINUE
C     AT THE HOLE OR AT ITS RIGHT EDGE
      X=(AUHOJ1-1.)*XSPAC1+(AI-AUHOJ1+1.)*XSPAC2
      NSUCRO=NUHOJ1*(NUVEJ1+NUVEJ2+NUVEJ3-2)+(I-NUHOJ1)*(NUVEJ1+NUVEJ3)
      GO TO 15
   11 CONTINUE
C     TO THE RIGHT OF HOLE
      NICK=NUHOJ1+NUHOJ2+NUHOJ3-3
      IF (J-NICK) 112,111,112
  111 X=HOLEN
      GO TO 113
  112 CONTINUE
      X=(AUHOJ1-1.)*XSPAC1+(AUHOJ2-1.)*XSPAC2+(AI-AUHOJ1-AUHOJ2+2.)*
```

```
     1XSPAC2
  115 CONTINUE
      NSUCRO=(NUVEJ+1)*(I-NUHOJ2+2)+(NUHOJ2-2)*(NUVEJ1+NUVEJ3)
   15 CONTINUE
C     Y COORDINATES
      NMAX=NUVEJ+1
      DO 27 JNDEX=1,NMAX
      J=JNDEX-1
      NVESPA=NUVEJ1-1
      AJ=J
      IF (J-NVESPA) 17,17,19
   17 CONTINUE
C     BELOW HOLE OR AT THE LOWER HOLE EDGE
      Y=AJ*YSPAC1
      GO TO 25
   19 CONTINUE
C     AT THE HOLE VERTICALLY OR OVER IT
      IF (I-NUHOJ1) 194,192,192
  192 CONTINUE
C     TO THE RIGHT OF LEFT EDGE OF HOLE
      IF (I-NUHOJ1-NUHOJ2+2) 193,194,194
  193 CONTINUE
C     AT THE HOLE VERTICALLY
      IF (J-NUVEJ1-NUVEJ3) 1935,30,30
 1935 CONTINUE
C     AT THE HOLE VERTICALLY BUT NOT AT UPPER EDGE OF PLATE
      Y=(AUVEJ1-1.)*YSPAC1+VELEN2+(AJ-AUVEJ1)*YSPAC3
      GO TO 25
  194 CONTINUE
C     OUTSIDE HOLE X-WISE AND OVER ITS LOWER EDGE
      NVESPA=NUVEJ1+NUVEJ2-1
      IF (J-NVESPA+1) 21,21,23
21    CONTINUE
C     AT THE HOLL Y-WISE,BUT OUTSIDE X-WISE
      Y=(AUVEJ1-1.)*YSPAC1+(AJ-AUVEJ1+1.)*YSPAC2
      GO TO 25
   23 CONTINUE
C     UPPER RIGHT CORNER
      Y=(AUVEJ1-1.)*YSPAC1+(AUVEJ2-1.)*YSPAC2+(AJ-AUVEJ2-AUVEJ1+2.)*
     1YSPAC3
   25 CONTINUE
C     FILL ARRAYS
      ERROR=ABS(Y-VELEN)
      PERER=(1./1000.)*VELEN
      IF (ERROR-PERER) 251,251,252
  251 Y=VELEN
  252 CONTINUE
      K=NSUCRO+J+1
      K1=K
      NUJO(K)=K1
      HOCOR(K)=X
      VECOR(K)=Y
   27 CONTINUE
   30 CONTINUE
      DO 31 J=1,NJ
```

```
      JORE(I,J)=0.
   31 CONTINUE
   41 CONTINUE
      RETURN
      END
```

```
**026
      SUBROUTINE MEINNH
      DYNAMIC ARRAY HOCOR,VECOR,NUJO,JOKENO,KINDJO,NUPLJO,NUMIJO,NUMEMB
      DYNAMIC ARRAY KINDME,NUJO1,HOCOR1,VECOR1
      COMMON HOLEN,VELEN,THICK,MANUJO,YOUNG,PRATIO,HOLEN1,HOLEN2,VELEN1
     1,VELEN2,EX,WHY,KIND1,EX2,WHY2,POLOM1,KILILO,LTYPE,NUHOJ1,NUVEJ1,
     2NJ,HOCOR(P),VECOR(P),NUJO(P),JOKENO(P),HOLEN3,VELEN3,NUHOJ2,NUHOJ3
     3,NUVEJ2,NUVEJ3,NUPLJO(P),NUMIJO(P),NUMEMB(P),KINDME(P),NUME,NUVEMB
     4,NUHOMB,NURIDI,NULEDI,NUPOLO,NUJO1(P),HOCOR1(P),VECOR1(P),NSIZE,
     5NULDS,ENDLD,LDMSTA,LDMEND,KINDJO(P),KIOUT,NLOTY
      EQUIVALENCE (NUHOJ1,NUHOJO),(NUVEJ1,NUVEJO)
      DO 4 I=1,NUME
      KINDME(I)=0
    4 CONTINUE
      NUJ=NUHOJO*NUVEJO-1
      K=0
      DO 20 I=1,NUJ
      IF (HOCOR(I)-HOCOR(I+1)) 10,5,10
    5 CONTINUE
      J=I-K
      NUMIJO(J)=NUJO(I)
      NUPLJO(J)=NUJO(I+1)
      NUMEMB(J)=J
      GO TO 20
   10 CONTINUE
      K=K+1
   20 CONTINUE
C     HORIZ.MEMBERS
      K=0
      NUJ=NUVEJO*(NUHOJO-1)
      DO 40 I=1,NUJ
      J=I+(NUVEJO-1)*NUHOJO
      NUMIJO(J)=NUJO(I)
      K=I+NUVEJO
      NUPLJO(J)=NUJO(K)
      NUMEMB(J)=J
   40 CONTINUE
C     DIAGONAL MEMBERS
      NUJ=NUVEJO*(NUHOJO-1)-1
      K=0
      DO 60 I=1,NUJ
      RATIO=VECOR(I)/VECOR(NUVEJO)
      IF (RATIO-1.) 45,50,45
   45 CONTINUE
C     THERE IS A DIAGONAL
      J=I+(NUVEJO-1)*NUHOJO+(NUHOJO-1)*NUVEJO-K
      NUMIJO(J)=NUJO(I)
      NUB=I+NUVEJO+1
      NUPLJO(J)=NUJO(NUB)
      NUMEMB(J)=J
      KINDME(J)=1
      GO TO 60
   50 CONTINUE
C     NO DIAGONAL(JOINT IS ON PERIMETER)
      K=K+1
```

```
   60 CONTINUE
      NUJ=NUVEJO*NUHOJO-1
      NUJ1=NUVEJO+1
      K=0
      DO 80 I=NUJ1,NUJ
      RATIO=VECOR(I)/VECOR(NUVEJO)
      IF (RATIO-1.) 65,70,65
   65 CONTINUE
C     THERE IS A DIAGONAL
      J=I-K+3*NUVEJO*NUHOJO-3*NUVEJO-2*NUHOJO+1
      NUMIJO(J)=NUJO(I)
      NUB=I-NUVEJO+1
      NUPLJO(J)=NUJO(NUB)
      NUMEMB(J)=J
      KINDME(J)=1
      GO TO 80
   70 CONTINUE
C     NO DIAGONAL
      K=K+1
   80 CONTINUE
      RETURN
      END
```

```
*AL 26
      SUBROUTINE MEINHO
      DYNAMIC ARRAY HOCOR,VECOR,NUJO,JORENO,KINDJO,NUPLJO,NUMIJO,NUMEMB
      DYNAMIC ARRAY KINDME,NUJO1,HOCOR1,VECOR1
      COMMON HOLEN,VELEN,THICK,MANUJO,YOUNG,PRATIO,HOLEN1,HOLEN2,VELEN1
     1,VELEN2,EX,WHY,KIND1,EX2,WHY2,POLOM1,KILILO,LTYPE,NUHOJ1,NUVEJ1,
     2NJ,HOCOR(P),VECOR(P),NUJO(P),JORENO(P),HOLEN3,VELEN3,NUHOJ2,NUHOJ3
     3,NUVEJ2,NUVEJ3,NUPLJO(P),NUMIJO(P),NUMEMB(P),KINDME(P),NUME,NUVEMB
     4,NUHOMB,NJRIDI,NULEJ1,NJPOLO,NUJO1(P),HOCOR1(P),VECOR1(P),NSIZE,
     5NULDS,ENDLD,LDMSTA,LDMEND,KINDJO(P),KIOUT,NLOTY
      NUVEJO=NUVEJ1+NUVEJ2+NUVEJ3-2
      NUHOJO=NUHOJ1+NUHOJ2+NUHOJ3-2
    3 FORMAT (3I5)
      DO 4 I=1,NUME
      KINDME(I)=0
    4 CONTINUE
C     VERTICAL MEMBERS
      NUJ=NJ-1
      K=0
      DO 20 I=1,NUJ
      NALPHA=NUVEJO*NUHOJ1+(NUVEJ1-1)
      NBETA=NALPHA+(NUHOJ2-3)*(NUVEJ1+NUVEJ3)+2
      IF (I-NALPHA) 12,12,11
   11 CONTINUE
      IF (I-NBETA)  13,12,12
   12 CONTINUE
C     REGULAR PROCEDURE
      IF (HOCOR(I)-HOCOR(I+1)) 122,121,122
  121 CONTINUE
      J=I-K
      NUMIJO(J)=NUJO(I)
      NUPLJO(J)=NUJO(I+1)
      NUMEMB(J)=J
      GO TO 20
  122 CONTINUE
      K=K+1
      GO TO 20
   13 CONTINUE
      IF (VECOR(I)-VECOR(NUVEJ1)) 131,122,131
  131 CONTINUE
      GO TO 12
   20 CONTINUE
C     HORIZ.MEMBERS
      NALPHA=NALPHA-(NUVEJO-1)
      NUJ=NJ-NUVEJO
      K=0
      DO 30 I=1,NUJ
      IF (I-NALPHA) 22,22,21
   21 CONTINUE
      IF (I-NBETA) 23,22,22
   22 CONTINUE
C     REGULAR PROCEDURE
      J=NUVEMB+I-K
      NUMIJO(J)=NUJO(I)
      NSUB=I+NUVEJO
```

```
      NUPLJO(J)=NUJO(NSUB)
      NUMEMB(J)=J
      GO TO 30
   23 CONTINUE
      IF (VECOR(I)-VECOR(NUVEJ1)) 233,233,231
  231 CONTINUE
      NUB=NUVEJ1+NUVEJ2-1
      IF (VECOR(I)-VECOR(NUB)) 232,233,233
  232 CONTINUE
      K=K+1
      GO TO 30
  233 CONTINUE
      J=NUVEMB+I-K
      NUP=I+NUVEJO-(NUVEJ2-2)
      NUMIJO(J)=NUJO(I)
      NUPLJO(J)=NUJO(NUP)
      NUMEMB(J)=J
      GO TO 30
   30 CONTINUE
C     RIGHT DIAGONAL MEMBERS
      NALPHA=NALPHA-1
      NUJ=NUJ-1
      K=0
      DO 40 I=1,NUJ
      IF (I-NALPHA) 32,32,31
   31 CONTINUE
      IF (I-NBETA) 33,32,32
   32 CONTINUE
C     REGULAR PROCEDURE
      NUP=I+NUVEJO+1
  323 CONTINUE
      IF (VECOR(I)-VECOR(NUVEJO)) 321,322,321
  321 CONTINUE
      J=NUVEMB+NUHOMB+I-K
      NUMIJO(J)=NUJO(I)
      NUPLJO(J)=NUJO(NUP)
      NUMEMB(J)=J
      KINDME(J)=1
      GO TO 40
  322 CONTINUE
      K=K+1
      GO TO 40
   33 CONTINUE
C     VICINITY OF HOLE
      IF (VECOR(I)-VECOR(NUVEJ1)) 332,331,331
  331 CONTINUE
      NUB=NUVEJ1+NUVEJ2-2
      IF (VECOR(I)-VECOR(NUB)) 322,322,332
  332 CONTINUE
      NUP=I+NUVEJO-NUVEJ2+3
      GO TO 323
   40 CONTINUE
C     LEFT DIAGONAL MEMBERS
      NALPHA=NUVEJO*NUHOJ1+(NUVEJ1-1)
      NBETA=NBETA+NUVEJO
```

```
      NUJ=NJ-1
      NUJ1=NUVEJO+1
      K=0
      DO 50 I=NUJ1,NUJ
      IF (I-NALPHA) 42,42,41
   41 CONTINUE
      IF(I-NBETA) 43,42,42
   42 CONTINUE
C     REGULAR PROCEDURE
      NUP=I-NUVEJO+1
  423 CONTINUE
      IF (VECOR(I)-VECOR(NUVEJO)) 421,422,421
  421 CONTINUE
      J=NUVEMB+NUHOMB+NURIDI-NUVEJO+I-K
      WRITE (6,3) I,J,K
      NUMIJO(J)=NUJO(I)
      NUPLJO(J)=NUJO(NUP)
      NUMEMB(J)=J
      KINDME(J)=1
      GO TO 50
  422 CONTINUE
      K=K+1
      GO TO 50
   43 CONTINUE
C     VICINITY OF HOLE
      IF (VECOR(I)-VECOR(NUVEJ1)) 432,431,431
  431 CONTINUE
      NUB=NUVEJ1+NUVEJ2-2
      IF (VECOR(I)-VECOR(NUB)) 422,422,432
  432 CONTINUE
      NUP=I-NUVEJO+NUVEJ2-1
      GO TO 423
   50 CONTINUE
      RETURN
      END
```

```
**026
      SUBROUTINE POBOC
      DYNAMIC ARRAY HOCOR,VECOR,NUJO,JORENU,KINDJO,NUPLJO,NUMIJO,NUMEMB
      DYNAMIC ARRAY KINDME,NUJO1,HOCOR1,VECOR1
      COMMON HOLEN,VELEN,THICK,MANUJO,YOUNG,PRATIO,HOLEN1,HOLEN2,VELEN1
     1,VELEN2,EX,WHY,KIND1,EX2,WHY2,POLOM1,KILILO,LTYPE,NUHOJ1,NUVEJ1,
     2NJ,HOCOR(P),VECOR(P),NUJO(P),JORENU(P),HOLEN3,VELEN3,NUHOJ2,NUHOJ3
     3,NUVEJ2,NUVEJ3,NUPLJO(P),NUMIJO(P),NUMEMB(P),KINDME(P),NUME,NUVEMB
     4,NUHOMB,NURIDI,NULEDI,NUPOLO,NUJO1(P),HOCOR1(P),VECOR1(P),NSIZE,
     5NULDS,ENDLD,LDMSTA,LDMEND,KINDJO(P),KIOUT,NLOTY
C     THIS SUBROUTINE TAKES EACH PARTCULAR POINT SUPPORT WITH
C     COORDINATES,EX(I)AND WHY(I) AND OF KIND,KIND1(I) AND MATCHES
C     IT WITH THE NEAREST GRID JOINT THE NUMBER OF WHICH IS
C     NUJO(ALPHA) WITH COORD. HOCOR(ALPHA) AND VECOR(ALPHA).
      DIST1=(EX    -HOCOR(1))**2+(WHY    -VECOR(1))**2
      NALPHA=1
      DO 100 J=2,NJ
      DIST=(EX    -HOCOR(J))**2+(WHY    -VECOR(J))**2
      IF (DIST1-DIST)10,10,30
   10 CONTINUE
C     DISTANCE FROM NEW POINT GREATER THAN OR EQUAL TO PREVIOUS
      GO TO 100
   30 CONTINUE
C     DISTANCE FROM NEW POINT LESS THAN PREVIOUS
      DIST1=DIST
      NALPHA=J
  100 CONTINUE
C     NEAREST JOINT FOUNT
      IF (KINDJO(NALPHA)-KIND1    ) 150,200,200
  150 KINDJO(NALPHA)=KIND1
  200 CONTINUE
      RETURN
      END
```

```
**026
      SUBROUTINE LIBOC
      DYNAMIC ARRAY HOCOR,VECOR,NUJO,JORENU,KINDJO,NUPLJO,NUMIJO,NUMEMB
      DYNAMIC ARRAY KINDME,NUJO1,HOCOR1,VECOR1
      COMMON HOLEN,VELEN,THICK,MANUJO,YOUNG,PRATIO,HOLEN1,HOLEN2,VELEN1
     1,VELEN2,EX,WHY,KIND1,EX2,WHY2,POLOM1,KILILO,LTYPE,NUHOJ1,NUVEJ1,
     2NJ,HOCOR(P),VECOR(P),NUJO(P),JORENU(P),HOLEN3,VELEN3,NUHOJ2,NUHOJ3
     3,NUVEJ2,NUVEJ3,NUPLJO(P),NUMIJO(P),NUMEMB(P),KINDME(P),NUME,NUVEMB
     4,NUHOMB,NURIDI,NULEDI,NUPOLO,NUJO1(P),HOCOR1(P),VECOR1(P),NSIZE,
     5NULDS,ENDLD,LDMSTA,LDMEND,KINDJO(P),KIOUT,NLOTY
      EQUIVALENCE (EX,EX1),(WHY,WHY1),(KIND1,KIND2)
C     THIS SUBROUTINE TAKES A SET OF LINE SUPPORTS AND APPROXIMATES
C     THEM BY A SET OF EQUIVALENT GRID JOINT SUPPORS
      DO 100 J=1,NJ
C     FIND EXACTLY MATCHING COOR.
      IF (EX1   -EX2   ) 20,10,20
   10 CONTINUE
C     X-COORD. MATCHES
      IF (EX1   -HOCOR(J)) 11,12,11
   11 CONTINUE
C     NOT A JOINT ON BOUNDARY
      GO TO 100
   12 CONTINUE
C     A JOINT ON BOUNDARY
      IF (WHY2   -VECOR(J))13,14,14
   13 CONTINUE
C     JOINT OUTSIDE Y-RANGE
      GO TO 100
   14 CONTINUE
C     JOINT PROBABLY INSIDE Y RANGE,CHECK Y1.
      IF (WHY1   -VECOR(J)) 16,16,15
   15 CONTINUE
C     JOINT OUTSIDE Y-RANGE
      GO TO 100
   16 CONTINUE
C     JOINT DEFINITELY WITHIN Y-RANGE
      JORENU(J)=JORENU(J)+1
      GO TO 80
   20 CONTINUE
C     Y-COORD. MATCHES
      IF (WHY1   -VECOR(J)) 21,22,21
   21 CONTINUE
C     NOT A JOINT ON BOUNDARY
      GO TO 100
   22 CONTINUE
C     A JOINT ON BOUNDARY
      IF (EX2   -HOCOR(J)) 23,24,24
   23 CONTINUE
C     JOINT OUTSIDE X-RANGE
      GO TO 100
   24 CONTINUE
C     JOINT PROBABLY INSIDE X-RANGE,CHECK X1
      IF (EX1   -HOCOR(J)) 26,26,25
   25 CONTINUE
C     JOINT OUTSIDE X-RANGE
      GO TO 100
   26 CONTINUE
      JORENU(J)=JORENU(J)+2
C     JOINT DEFINITELY WITHIN X-RANGE
```

```
   80 IF (KINDJO(J)-KIND2    ) 90,100,100
   90 KINDJO(J)=KIND2
  100 CONTINUE
C     PICK ANOTHER JOINT
  200 CONTINUE
C     PICK ANOTHER BOUNDARY CONDITION
      RETURN
      END
```

```
**026
      SUBROUTINE POINLO
      DYNAMIC ARRAY HOCOR,VECOR,NUJO,JORENU,KINDJO,NUPLJO,NUMIJO,NUMEMB
      DYNAMIC ARRAY KINDME,NUJO1,HOCOR1,VECOR1
      COMMON HOLEN,VELEN,THICK,MANUJO,YOUNG,PRATIO,HOLEN1,HOLEN2,VELEN1
     1,VELEN2,EX,WHY,KIND1,EX2,WHY2,POLOM1,KILILO,LTYPE,NUHOJ1,NUVEJ1,
     2NJ,HOCOR(P),VECOR(P),NUJO(P),JORENU(P),HOLEN3,VELEN3,NUHOJ2,NUHOJ3
     3,NUVEJ2,NUVEJ3,NUPLJO(P),NUMIJO(P),NUMEMB(P),KINDME(P),NUME,NUVEMB
     4,NUHOMB,NURIDI,NULEDI,NUPOLO,NUJO1(P),HOCOR1(P),VECOR1(P),NSIZE,
     5NULDS,ENDLD,LDMSTA,LDMEND,KINDJO(P),KIOUT,NLOTY
      EQUIVALENCE )EX,EXLOAD*,)WHY,WYLOAD*
C     THIS SUBROUTINE TAKES EACH POINT LOAD AND FINDS THE NEAREST NODE
C     OF THE GRID WHICH IS NOT A SUPPORT,TO BE CONCIDERED AS POINT OF
C     APPLICATION OF LOAD.
   70 FORMAT (T8,12H JOINT LOADS)
   75 FORMAT (T8,I4,9H FORCE Z ,F15.3)
   80 FORMAT (T8,I4,10H MOMENT X ,F15.3)
   83 FORMAT (T8,11H LIST LOADS ,I4)
   85 FORMAT (T8,I4,10H MOMENT Y ,F15.3)
   86 FORMAT (T8,20H LIST DISPLACEMENTS ,I4)
C     FORM AN ARRAY OF NON-SUPPORT JOINTS
      CALL NOSUJO
      IF (NUPOLO-1) 22,21,22
   21 CONTINUE
      WRITE (7,70)
   22 DIST1=(EXLOAD-HOCOR1(1))**2+(WYLOAD-VECOR1(1))**2
      NALPHA=1
      DO 100 J=2,NSIZE
      DIST =(EXLOAD-HOCOR1(J))**2+(WYLOAD-VECOR1(J))**2
      IF (DIST1-DIST) 100,100,30
   30 DIST1=DIST
      NALPHA=J
  100 CONTINUE
C     NEAREST JOINT FOUND
      GO TO (101,101,103,103,103,106),KIOUT
  103 CONTINUE
      NUJO9=NUJO1(NALPHA)
      WRITE (7,83) NUJO9
      GO TO 200
  106 CONTINUE
      NUJO9=NUJO1(NALPHA)
      WRITE (7,86) NUJO9
      GO TO 200
  101 CONTINUE
      IF (NLOTY-1) 92,91,92
   91 CONTINUE
C     NLOTY CONTAINS AN 1.LOAD IS A Z FORCE
      NUJO9=NUJO1(NALPHA)
      WRITE (7,75) NUJO9,POLOM1
      GO TO 200
   92 CONTINUE
C     NLOTY DOESN'T CONTAIN AN 1.CHECK IF 2 DOES.
      IF (NLOTY-2) 94,93,94
   93 CONTINUE
C     NLOTY CONTAINS A 2.
      NUJO9=NUJO1(NALPHA)
      WRITE (7,80) NUJO9,POLOM1
      GO TO 200
   94 CONTINUE
```

```
C     NLOTY CONTAINS A 3.
      NUJO9=NUJO1(NALPHA)
      WRITE (7,85) NUJO9,POLOM1
  200 CONTINUE
      NUPOLO=NUPOLO+1
      RETURN
      END
```

```
**026
      SUBROUTINE NOSUJO
      DYNAMIC ARRAY HOCOR,VECOR,NUJO,JORENU,KINDJO,NUPLJO,NUMIJO,NUMEMB
      DYNAMIC ARRAY KINDME,NUJO1,HOCOR1,VECOR1
      COMMON HOLEN,VELEN,THICK,MANUJO,YOUNG,PRATIO,HOLEN1,HOLEN2,VELEN1
     1,VELEN2,EX,WHY,KIND1,EX2,WHY2,POLOM1,KILILO,LTYPE,NUHOJ1,NUVEJ1,
     2NJ,HOCOR(P),VECOR(P),NUJO(P),JORENU(P),HOLEN3,VELEN3,NUHOJ2,NUHOJ3
     3,NUVEJ2,NUVEJ3,NUPLJO(P),NUMIJO(P),NUMEMB(P),KINDME(P),NUME,NUVEMB
     4,NUHOMB,NURIDI,NULEDI,NUPOLO,NUJO1(P),HOCOR1(P),VECOR1(P),NSIZE,
     5NULDS,ENDLD,LDMSTA,LDMEND,KINDJO(P),KIOUT,NLOTY
      K=0
      DO 50 I=1,NJ
      IF (KINDJO(I)-0) 10,5,10
    5 CONTINUE
C     NOT A SUPPORT
      J=I-K
      NUJO1(J)=NUJO(I)
      HOCOR1(J)=HOCOR(I)
      VECOR1(J)=VECOR(I)
      GO TO 50
   10 CONTINUE
C     A SUPPORT JOINT SKIP.
      K=K+1
   50 CONTINUE
      NSIZE=J
      RETURN
      END
```

```
**026
**OBJECT
      SUBROUTINE JOINRE
      DYNAMIC ARRAY HOCOR,VECOR,NUJO,JORENU,KINDJO,NUPLJO,NUMIJO,NUMEMB
      DYNAMIC ARRAY KINDME,NUJO1,HOCOR1,VECOR1
      DYNAMIC ARRAY NU
      COMMON HOLEN,VELEN,THICK,MANUJO,YOUNG,PRATIO,HOLEN1,HOLEN2,VELEN1
     1,VELEN2,EX,WHY,KIND1,EX2,WHY2,POLOM1,KILILO,LTYPE,NUHOJ1,NUVEJ1,
     2NJ,HOCOR(P),VECOR(P),NUJO(P),JORENU(P),HOLEN3,VELEN3,NUHOJ2,NUHOJ3
     3,NUVEJ2,NUVEJ3,NUPLJO(P),NUMIJO(P),NUMEMB(P),KINDME(P),NUME,NUVEMB
     4,NUHOMB,NURIDI,NULEDI,NUPOLO,NUJO1(P),HOCOR1(P),VECOR1(P),NSIZE,
     5NULDS,ENDLD,LDMSTA,LDMEND,KINDJO(P),KIOUT,NLOTY
    3 FORMAT (T8,20H TYPE PLANE GRID XY )
    4 FORMAT (T8,15H CONSTANTS E , ,F11.3,5H,ALL )
    5 FORMAT (T8,15H CONSTANTS G , ,F11.3,5H,ALL )
    6 FORMAT (T8,18HJOINT COORDINATES )
    8 FORMAT (T8,I4,2F10.4,9H SUPPORT )
   10 FORMAT (T8,I4,2F10.4)
   12 FORMAT (T8,15HJOINT RELEASES )
   14 FORMAT (T8,I4,12H MOMENT X Y )
   15 FORMAT (T8,I4,10H MOMENT X )
   16 FORMAT (T8,I4,10H MOMENT Y )
   17 FORMAT (T8,19H UNITS,INCHES,KIPS )
      WRITE (7,3)
      WRITE (7,4) YOUNG
      G=YOUNG/(2.*(1.+PRATIO))
      WRITE (7,5) G
      WRITE (7,17)
      WRITE (7,6)
      K=1
      DEFINE NU,NJ
      DO 100 I=1,NJ
      KI=KINDJO(I)+1
      GO TO (80,50,70,50),KI
   50 CONTINUE
C     JOINT IS HINGED,RELEASE MOMENTS
      NU(K)=NUJO(I)
      K=K+1
      NUJ9=NUJO(I)
      HOCO9=HOCOR(I)
      VECO9=VECOR(I)
      WRITE (7,8) NUJ9,HOCO9,VECO9
      GO TO 100
   70 CONTIPUE
C     JOINT IS FIXED
      NUJ9=NUJO(I)
      HOCO9=HOCOR(I)
      VECO9=VECOR(I)
      WRITE (7,8) NUJ9,HOCO9,VECO9
      GO TO 100
   80 CONTINUE
C     JOINT IS FREE
      NUJ9=NUJO(I)
      HOCO9=HOCOR(I)
      VECO9=VECOR(I)
      WRITE (7,10) NUJ9,HOCO9,VECO9
  100 CONTINUE
      IF (K-1) 110,110,120
  110 CONTINUE
```

```
C       THERE ARE NO JOINT RELEASES
        GO TO 200
    120 CONTINUE
C       THERE ARE K-1 JOINT RELEASES
        WRITE (7,12)
        L=K-1
        DO 150 K=1,L
        NOCO=QQFDI1(NU,K)
        KI=QQFDI1(JORENU,NOCO)+1
        GO TO (140,130,130),KI
    130 CONTINUE
C       A LINE SUPPORT.RELEASE ONLY ONF MOMENT
        NALPHA=NU(K)
        JOR=JORENU(NALPHA)
        GO TO (131,132,140),JOR
    131 N9=NU(K)
        WRITE (7,16) N9
        GO TO 150
    132 N9=NU(K)
        WRITE (7,15) N9
        GO TO 150
    140 N9=NU(K)
        WRITE (7,14) N9
    150 CONTINUE
    200 CONTINUE
    201 FORMAT (T8,19H MEMBER INCIDENCES )
        WRITE (7,201)
    202 FORMAT (T8,4I4)
        DO 300 I=1,NUME
        NUMEM9=NUMEMB(I)
        NUPLJ9=NUPLJO(I)
        NUMIJ9=NUMIJO(I)
        WRITE (7,202) NUMEM9,NUMIJ9,NUPLJ9
    300 CONTINUE
        DESTROY NU
        RETURN
        END
```

```
**026
      SUBROUTINE MEMPRO
      DYNAMIC ARRAY HOCOR,VECOR,NUJO,JORENU,KINDJO,NUPLJO,NUMIJO,NUMEMB
      DYNAMIC ARRAY KINDME,NUJO1,HOCOR1,VECOR1
      DYNAMIC ARRAY MEMNU,REINER,TOINER
      COMMON HOLEN,VELEN,THICK,MANUJO,YOUNG,PRATIO,HOLEN1,HOLEN2,VELEN1
     1,VELEN2,EX,WHY,KIND1,EX2,WHY2,POLOM1,KILILO,LTYPE,NUHOJ1,NUVEJ1,
     2NJ,HOCOR(P),VECOR(P),NUJO(P),JORENU(P),HOLEN3,VELEN3,NUHOJ2,NUHOJ3
     3,NUVEJ2,NUVEJ3,NUPLJO(P),NUMIJO(P),NUMEMB(P),KINDME(P),NUME,NUVEMB
     4,NUHOMB,NURIDI,NULEDI,NUPOLO,NUJO1(P),HOCOR1(P),VECOR1(P),NSIZE,
     5NULDS,ENDLD,LDMSTA,LDMEND,KINDJO(P),KIOUT,NLOTY
      DEFINE MEMNU ,NUME
      DEFINE REINER,NUME
      DEFINE TOINER,NUME
    4 FORMAT (T8,28HMEMBER PROPERTIES,PRISMATIC )
    6 FORMAT (T8,I4,4H IY ,F15.5,4H IX ,F15.5)
    8 FORMAT (T8,13H LOADING '1' )
      FACTOR=(THICK**3)/(24.*(1.-PRATIO**2))
      WRITE (7,4)
      DO 10 I=1,NUME
      REINER(I)=0
   10 TOINER(I)=0
C     ARRAYS SET EQUAL TO ZERO
      I1=NUHOMB+NUVEMB
      I2=I1+NURIDI
      DO 200 I=I1,I2
      N=0
      DO 100 J=1,I1
      NALPHA=NUMIJO(I)
      NBETA=NUPLJO(I)
      NALPH=NUMIJO(J)
      NBET=NUPLJO(J)
      IF (NALPHA-NALPH) 100,50,100
   50 N=N+1
      MEMNU(N)=NUMEMB(J)
  100 CONTINUE
      NAO=N+1
      GO TO (200,200,110),NAO
  110 CONTINUE
      DO 120 K=1,I1
      NALPH=NUMIJO(K)
      NBET=NUPLJO(K)
      IF (NBETA-NBET) 120,111,120
  111 N=N+1
      MEMNU(N)=NUMEMB(K)
  120 CONTINUE
      DO 130 K=I2,NUME
      NALPH=NUMIJO(K)
      NBET=NUPLJO(K)
      NCAT=MEMNU(1)
      IF (NBET-NUPLJO(NCAT)) 130,121,130
  121 N=N+1
      MEMNU(N)=NUMEMB(K)
  130 CONTINUE
      N=N+1
      MEMNU(N)=NUMEMB(I)
C     SIX BEAMS OF PANNEL FOUND
      VESIDE=VECOR(NBETA)-VECOR(NALPHA)
      HOSIDE=HOCOR(NBETA)-HOCOR(NALPHA)
```

```
      ADIAG=SQRT(VESIDE**2+HOSIDE**2)
      HOVERA=HOSIDE/VESIDE
      DIVERA=ADIAG/VESIDE
      DO 180 K=1,N
      NGAMMA=MEMNU(K)
      GO TO (161,162,161,162,163,163),K
  161 COEF1=(HOVERA-PRATIO/HOVERA)*VESIDE
      COEF2=(2*HOVERA*(1.+PRATIO)*(1.-3.*PRATIO))*VESIDE
      GO TO 170
  162 COEF1=(1.-(HOVERA**2)*PRATIO)*VESIDE
      COEF2=(2.*(1.+PRATIO)*(1.-3.*PRATIO))*VESIDE
      GO TO 170
  163 COEF1=(PRATIO/HOVERA)*(DIVERA**3)*VESIDE
      COEF2=0
  170 REINER(NGAMMA)=COEF1*FACTOR+REINER(NGAMMA)
      TOINER(NGAMMA)=COEF2*FACTOR+TOINER(NGAMMA)
  180 CONTINUE
  200 CONTINUE
      DO 300 K=1,NUME
      NUMEM9=NUMEMB(K)
      REINE9=REINER(K)
      TOINE9=TOINER(K)
      WRITE (7,6) NUMEM9,REINE9,TOINE9
  300 CONTINUE
      DESTROY MEMNU
      DESTROY REINER
      DESTROY TOINER
      WRITE (7,8)
      RETURN
      END
```

```
**026
      SUBROUTINE LINELO
      DYNAMIC ARRAY HOCOR,VECOR,NUJO,JORENU,KINDJO,NUPLJO,NUMIJO,NUMEMB
      DYNAMIC ARRAY KINDME,NUJO1,HOCOR1,VECOR1
      DYNAMIC ARRAY XLOCAL,YLOCAL,XGLLD,YGLLD,NULOME,KINDLO,NULOJO
      COMMON HOLEN,VELEN,THICK,MANUJO,YOUNG,PRATIO,HOLEN1,HOLEN2,VELEN1
     1,VELEN2,EX,WHY,KIND1,EX2,WHY2,POLOM1,KILILO,LTYPE,NUHOJ1,NUVEJ1,
     2NJ,HOCOR(P),VECOR(P),NUJO(P),JORENU(P),HOLEN3,VELEN3,NUHOJ2,NUHOJ3
     3,NUVEJ2,NUVEJ3,NUPLJO(P),NUMIJO(P),NUMEMB(P),KINDME(P),NUME,NUVEMB
     4,NUHOMB,NURIDI,NULEDI,NUPOLO,NUJO1(P),HOCOR1(P),VECOR1(P),NSIZE,
     5NULDS,ENDLD,LDMSTA,LDMEND,KINDJO(P),KIOUT,NLOTY
      EQUIVALENCE (EX,XA),(EX2,XB),(WHY,YA),(WHY2,YB),(STALD,POLOM1)
      DEFINE XLOCAL ,NUVEMB+NUHOMB
      DEFINE YLOCAL ,NUVEMB+NUHOMB
      DEFINE XGLLD  ,NUVEMB+NUHOMB
      DEFINE YGLLD  ,NUVEMB+NUHOMB
      DEFINE NULOME ,NUVEMB+NUHOMB
      DEFINE NULOJO,NUVEMB+NUHOMB
      DEFINE KINDLO,NUVEMB+NUHOMB
      K=0
      N=0
      IF (XA-XB) 40,41,40
   40 CONTINUE
      IF (YA-YB) 49,42,49
   41 CONTINUE
C     LOADING LINE PARALLEL TO Y- AXIS
      DO 411 I=1,NJ
      IF (XA-HOCOR(I)) 411,412,411
  411 CONTINUE
C     LOADING LINE DOESN'T COINCIDE WITH A SET OF MEMBERS
      K=0
      N=0
      GO TO 100
  412 CONTINUE
C     LOADING LINE COINCIDES WITH A SET OF MEMBERS
      K=0
      DO 419 J=1,NUVEMB
      NALPHA=NUMIJO(J)
      NBETA=NUPLJO(J)
      IF (HOCOR(NALPHA)-XA) 419,413,419
  413 CONTINUE
      IF (VECOR(NBETA)-YA) 419,419,414
  414 CONTINUE
      IF (VECOR(NALPHA)-YB) 415,419,419
  415 CONTINUE
      IF (VECOR(NALPHA)-YA) 416,416,417
  416 CONTINUE
      K=K+1
      XLOCAL(K)=YA-VECOR(NALPHA)
      XGLLD(K)=YA
      GO TO 418
  417 CONTINUE
      K=K+1
      XLOCAL(K)=0
      XGLLD(K)=VECOR(NALPHA)
  418 CONTINUE
      NULOME(K)=NUMEMB(J)
      KINDLO(K)=2
      IF (VECOR(NBETA)-YB) 4161,4161,4162
```

```
 4161 CONTINUE
      YGLLD(K)=VECOR(NBETA)
      YLOCAL(K)=VECOR(NBETA)-VECOR(NALPHA)
      GO TO 419
 4162 CONTINUE
      YGLLD(K)=YB
      YLOCAL(K)=YB-VECOR(NALPHA)
  419 CONTINUE
      GO TO 200
   42 CONTINUE
C     LOADING LINE  PARALLEL TO X-AXIS
      DO 421 I=1,NJ
      IF (YA-VECOR(I)) 421,422,421
  421 CONTINUE
      GO TO 49
  422 CONTINUE
C     LOADIN LINE COINCIDES WITH A SET OF MEMBERS
      NUV=NUVEMB+1
      NUVE=NUVEMB+NUHOMB
      DO 429 J=NUV,NUVE
      NALPHA=NUMIJO(J)
      NBETA=NUPLJO(J)
      IF (VECOR(NALPHA)-YA) 429,423,429
  423 CONTINUE
      IF (HOCOR(NBETA)-XA) 429,429,424
  424 CONTINUE
      IF (HOCOR(NALPHA)-XB) 425,429,429
  425 CONTINUE
      IF (HOCOR(NALPHA)-XA) 426,426,427
  426 CONTINUE
      K=K+1
      XLOCAL(K)=XA-HOCOR(NALPHA)
      XGLLD(K)=XA
      GO TO 428
  427 CONTINUE
      K=K+1
      XLOCAL(K)=0
      XGLLD(K)=HOCOR(NALPHA)
  428 CONTINUE
      NULOME(K)=NUMEMB(J)
      KINDLO(K)=2
      IF (HOCOR(NBETA)-XB) 4261,4261,4262
 4261 CONTINUE
      YGLLD(K)=HOCOR(NBETA)
      YLOCAL(K)=HOCOR(NBETA)-HOCOR(NALPHA)
      GO TO 429
 4262 CONTINUE
      YGLLD(K)=XB
      YLOCAL(K)=XB-HOCOR(NALPHA)
  429 CONTINUE
      GO TO 200
   49 CONTINUE
C     LOADING LINE NOT OF THE FORM X=CONST.OR Y=CONST.
      N=0
      K=0
      COEF=(YB-YA)/(XB-XA)
      DO 100 I=1,NUVEMB
      NALPHA=NUMIJO(I)
      NBETA=NUPLJO(I)
```

```
      IF (N-0) 494,493,494
  494 CONTINUE
      DO 493 J=1,N
      IF (NALPHA-NULOJO(N))  493,100,493
  493 CONTINUE
      Y=YA+COEFF*(HOCOR(NALPHA)-XA)
      TEDPL=ABS(VECOR(NBETA)-Y)
      TED=(VECOR(NBETA)-VECOR(NALPHA))/1000.
      ABSY=ABS(Y)
      IF (VECOR(NALPHA)-0) 491,496,491
  496 CONTINUE
      IF (ABSY-TED) 57,57,491
  491 CONTINUE
      IF (TEDPL-TED) 56,56,492
  492 CONTINUE
      IF (XA-HOCOR(NALPHA)) 50,50,100
   50 CONTINUE
      IF (XB- HOCOR(NALPHA)) 100,51,51
   51 CONTINUE
C     INTERSECTION POSSIBLE BUT NOT CERTAIN
      IF (VECOR(NALPHA)-Y) 52,100,100
   52 CONTINUE
C     INTERSECTION STILL POSSIBLE
      IF (VECOR(NBETA)-Y) 100,56,55
   55 CONTINUE
C     INTERSECTION CERTAIN
      K=K+1
      NULOME(K)=NUMEMB(I)
      XLOCAL(K)=Y-VECOR(NALPHA)
      XGLLD(K)=HOCOR(NALPHA)
      YGLLD(K)=Y
      KINDLO(K)=0
      GO TO 100
   56 CONTINUE
C     PLUS JOINT OF MEMBER LIES ON LOAD LINE
      IF (KINDJO(NBETA)-1) 561,562,562
  561 CONTINUE
C     JOINT NOT A SUPPORT
      K=K+1
      NULOME(K)=NBETA
      XGLLD(K)=HOCOR(NBETA)
      YGLLD(K)=VECOR(NBETA)
      KINDLO(K)=1
      N=N+1
      NULOJO(N)=NBETA
      IF (YA-YB) 100,200,100
  562 N=N+1
      NULOJO(N)=NBETA
      GO TO 100
   57 CONTINUE
C     MINUS JOINT LIES ON LOAD LINE
      IF (KINDJO(NALPHA)-1) 571,572,572
  571 CONTINUE
C     JOINT NOT A SUPPORT
      K=K+1
      NULOME(K)=NALPHA
      XGLLD(K)=HOCOR(NALPHA)
      YGLLD(K)=VECOR(NALPHA)
      KINDLO(K)=1
```

```
      N=N+1
      NULOJO(N)=NALPHA
      IF (YA-YB) 100,200,100
  572 CONTINUE
      N=N+1
      NULOJO(N)=NALPHA
  100 CONTINUE
C     ALL INTERSECTION POINTS WITH VERTICALMEMBERS FOUNT.THERE ARE ALSO
C     N JOINT INTERSECTIONS
C     HORIZONTAL MEMBERS
      DO 200 INDEX=1,NUHOMB
      I=INDEX+NUVEMB
      NALPHA=NUMIJO(I)
      NBETA=NUPLJO(I)
      IF (N-0) 130,137,130
  130 CONTINUE
C     THE LIN  OF THE LOADING PASSES THRU SOME JOINTS
      DO 137 J=1,N
      IF (NALPHA-NULOJO(J)) 135,200,135
  135 CONTINUE
      IF (NBETA-NULOJO(J)) 137,200,137
  137 CONTINUE
C     MEMBER I DOESNOT PASS THRU A LOADED JOINT
      IF (YA-YB) 138,200,139
  138 CONTINUE
      IF (YB-VECOR(NALPHA)) 200,1381,1381
 1381 CONTINUE
      IF (YA-VECOR(NALPHA)) 151,151,200
  139 CONTINUE
      IF (YA-VECOR(NALPHA)) 200,1391,1391
 1391 CONTINUE
      IF(YB-VECOR(NALPHA)) 151,151,200
  151 CONTINUE
C     INTERSECTION POSSIBLE BUT NOT CERTAIN
      X=XA+((VECOR(NALPHA)-YA)*(XB-XA))/(YB-YA)
      IF (HOCOR(NALPHA)-X) 152,200,200
  152 CONTINUE
C     INTERSECTION STILL POSSIBLE
      IF (HOCOR(NBETA)-X) 200,200,155
  155 CONTINUE
C     INTERSECTION CERTAIN
      K=K+1
      NULOME(K)=NUMEMB(I)
      YGLLD(K)=VECOR(NALPHA)
      XGLLD(K)=X
      XLOCAL(K)=X-HOCOR(NALPHA)
      KINDLO(K)=0
  200 CONTINUE
      NULDS=K
      CALL LINE ($KINDLO,$XGLLD,$YGLLD,$NULOME,$XLOCAL,$YLOCAL)
      DESTROY XLOCAL
      DESTROY YLOCAL
      DESTROY XGLLD
      DESTROY YGLLD
      DESTROY NULOME
      DESTROY KINDLO
      DESTROY NULOJO
      RETURN
      END
```

```
**026
      SUBROUTINE LINE(KINDLO,XGLLD,YGLLD,NULOME,XLOCAL,YLOCAL)
      DYNAMIC ARRAY HOCOR,VECOR,NUJO,JORENU,KINDJO,NUPLJO,NUMIJO,NUMEMB
      DYNAMIC ARRAY KINDME,NUJO1,HOCOR1,VECOR1
      DYNAMIC ARRAY XLOCAL,YLOCAL,XGLLD,YGLLD,NULOME,KINDLO
      DYNAMIC ARRAY RADIUS,STALME,P,JOINT
      COMMON HOLEN,VELEN,THICK,MANUJO,YOUNG,PRATIO,HOLEN1,HOLEN2,VELEN1
     1,VELEN2,EX,WHY,KIND1,EX2,WHY2,POLOM1,KILILO,LTYPE,NUHOJ1,NUVEJ1,
     2NJ,HOCOR(P),VECOR(P),NUJO(P),JORENU(P),HOLEN3,VELEN3,NUHOJ2,NUHOJ3
     3,NUVEJ2,NUVEJ3,NUPLJO(P),NUMIJO(P),NUMEMB(P),KINDME(P),NUME,NUVEMB
     4,NUHOMB,NURIDI,NULEDI,NUPOLO,NUJO1(P),HOCOR1(P),VECOR1(P),NSIZE,
     5NULDS,ENDLD,LDMSTA,LDMEND,KINDJO(P),KIOUT,NLOTY
      EQUIVALENCE (EX,XA),(EX2,XB),(WHY,YA),(WHY2,YB),(STALD,POLOM1)
C     THIS SUBROUTINE IS A LOGICAL CONTINUATION OF LINELO.IT WAS FORMED
C     AS A SEPARATE PROGRAM SINCE PRECOMPILER AT THIS TIME CANNOT ACCEPT
C     SUCH A LARGE NUMBER OF SYMBOLS
C     SHORT LIST OF POINTS  ACCORDING TO THEIR DISTANCE FROM THE
C     LEADING EDGE OF THE LINE LOAD
      DEFINE RADIUS ,NUVEMB+NUHOMB
      DEFINE STALME ,NUVEMB+NUHOMB
      DEFINE P      ,NUVEMB+NUHOMB
      DEFINE JOINT  ,NUVEMB+NUHOMB
   10 FORMAT (T8,12H JOINT LOADS)
   11 FORMAT (T8,I4,9H FORCE Z ,F15.3)
   12 FORMAT (T8,I4,10H MOMENT X ,F15.3)
   13 FORMAT (T8,I4,10H MOMENT Y ,F15.3)
   14 FORMAT(T8,13H MEMBER LOADS)
   15 FORMAT (T8,I4,23H FORCE Z CONCENTRATED P,F15.3,2H L,F15.3)
   16 FORMAT (T8,I4,24H MOMENT X CONCENTRATED P,F15.3,2H L,F15.3)
   17 FORMAT (T8,I4,24H MOMENT Y CONCENTRATED P,F15.3,2H L,F15.3)
   18 FORMAT (T8,I4,19H FORCE Z UNIFORM W ,F15.3,4H LA ,F15.3,3H - /
     1T8,4H LB ,F15.3)
   19 FORMAT (T8,I4,20H MOMENT X UNIFORM W ,F15.3,4H LA ,F15.3,3H - /
     1T8,4H LB ,F15.3)
   20 FORMAT (T8,I4,20H MOMENT Y UNIFORM W ,F15.3,4H LA ,F15.3,3H - /
     1T8,4H LB ,F15.3)
   21 FORMAT (T8,I4,19H FORCE Z LINEAR WA ,F15.3,4H WB ,F15.3,3H - /
     1T8,4H LA ,F15.3,4H LB ,F15.3)
   22 FORMAT (T8,I4,20H MOMENT X LINEAR WA ,F15.3,4H WB ,F15.3,3H - /
     1T8,4H LA ,F15.3,4H LB ,F15.3)
   23 FORMAT (T8,I4,20H MOMENT Y LINEAR WA ,F15.3,4H WB ,F15.3,3H - /
     1T8,4H LA ,F15.3,4H LB ,F15.3)
   24 FORMAT (T8,11H LIST LOADS ,I4)
   25 FORMAT (T8,20H LIST DISPLACEMENTS ,I4)
      DO 201 I=1,NULDS
      IF (KINDLO(I)-1) 2009,2009,2006
 2006 CONTINUE
      IF (XA-XB) 2008,2007,2008
 2007 CONTINUE
      RADIUS(I)=XGLLD(I)-YA
      GO TO 2004
 2008 CONTINUE
      RADIUS(I)=XGLLD(I)-XA
      GO TO 2004
 2009 CONTINUE
      RADIUS(I)=SQRT((XGLLD(I)-XA)**2+(YGLLD(I)-YA)**2)
 2004 CONTINUE
      IF (LTYPE-1) 2002,2003,2002
 2002 CONTINUE
```

```
C       INITIAL LOADING UNIFORM
        STALME(I)=STALD
        GO TO 201
 2003 CONTINUE
        AEN=SQRT((XB-XA)**2+(YB-YA)**2)
C       INITIAL LOADING IS LINEAR
        STALME(I)=STALD+(RADIUS(I)/AEN)*(ENDLD-STALD)
  201 CONTINUE
        IF (KINDLO(I)-1) 2005,2005,350
 2005 CONTINUE
        DO 210 J=1,NULDS
        DO 210 L=2,NULDS
        I=L-1
        IF (RADIUS(L)-RADIUS(I)) 202,210,210
  202 CONTINUE
        NULOM=NULOME(I)
        XGLL=XGLLD(I)
        YGLL=YGLLD(I)
        XLOCA=XLOCAL(I)
        KINDL=KINDLO(I)
        RADIU=RADIUS(I)
        STALM=STALME(I)
        NULOME(I)=NULOME(L)
        XGLLD(I)=XGLLD(L)
        YGLLD(I)=YGLLD(L)
        XLOCAL(I)=XLOCAL(L)
        KINDLO(I)=KINDLO(L)
        RADIUS(I)=RADIUS(L)
        STALME(I)=STALME(L)
        NULOME(L)=NULOM
        XGLLD(L)=XGLL
        YGLLD(L)=YGLL
        XLOCAL(L)=XLOCA
        KINDLO(L)=KINDL
        RADIUS(L)=RADIU
        STALME(L)=STALM
  210 CONTINUE
C       LIST SHORTED
 2101 CONTINUE
        GO TO (2102,2102,2102,2103,2102,2102,2103),KIOUT
 2103 CONTINUE
        DO 2104 I=1,NULDS
        JUNK=KINDLO(I)+1
        GO TO (2105,2106,2107),JUNK
 2105 CONTINUE
C       LINE INTERSECTS A MEMBER
        MEMBER=NULOME(I)
        NGAMMA=NUMIJO(MEMBER)
        NDELTA=NUPLJO(MEMBER)
        RLEME=SQRT((HOCOR(NDELTA)-HOCOR(NGAMMA))**2+(VECOR(NDELTA)-VECOR(
       1NGAMMA))**2)
        IF (XLOCAL(I)-RLEME/2.) 2108,2108,2109
 2108 JOINT(I)=NUJO(NGAMMA)
        NADD=0
        GO TO 2104
 2109 JOINT(I)=NUJO(NDELTA)
        NADD=0
        GO TO 2104
 2106 CONTINUE
```

```
      JOINT(I)=NULOME(I)
      NADD=0
      GO TO 2104
 2107 CONTINUE
      MEMBER=NULOME(I)
      NGAMMA=NUMIJO(MEMBER)
      NDELTA=NUPLJO(MEMBER)
      JOINT(I)=NUJO(NGAMMA)
      JOINT(I+1)=NUJO(NGAMMA)
      NADD=1
 2104 CONTINUE
      NUL=NULDS+NADD
      IF (KIOUT-4) 2110,2111,2110
 2110 JOIN9=JOINT(1)
      WRITE (7,25) JOIN9
      DO 2112 I=2,NUL
      IF (JOINT(I)-JOINT(I-1)) 2113,2112,2113
 2113 JOIN9=JOINT(I)
      WRITE (7,25) JOIN9
 2112 CONTINUE
      GO TO 360
 2111 JOIN9=JOINT(1)
      WRITE (7,24) JOIN9
      DO 2114 I=2,NUL
      IF (JOINT(I)-JOINT(I-1)) 2115,2114,2115
 2115 JOIN9=JOINT(I)
      WRITE (7,24) JOIN9
 2114 CONTINUE
      GO TO 360
 2102 CONTINUE
      IF (NULDS-1) 212,211,212
  211 CONTINUE
C     ONLY ONE CONCENTRATED LOAD
      FACTOR=SQRT((XA-XB)**2+(YA-YB)**2)
      P(1)=FACTOR*((STALD+ENDLD)/2.)
      GO TO 300
  212 CONTINUE
C     MORE THAN ONE CONCENTRATED LOADS
      FACTOR=SQRT((XA-XGLLD(1))**2+(YA-YGLLD(1))**2)
      P(1)=FACTOR*((STALD +STALME(1))/2.)
      DO 213 L=2,NULDS
      I=L-1
      FACTOR=RADIUS(L)-RADIUS(I)
      C=FACTOR*(2.*ABS(STALME(L))+ABS(STALME(I)))/(3.*(ABS(STALME(L))+
     1ABS(STALME(I))))
      P(L)=C*((STALME(L)+STALME(I))/2.)
      P(I)=P(I)+FACTOR*((STALME(L)+STALME(I))/2.)-P(L)
  213 CONTINUE
      FACTOR=SQRT((YB-YGLLD(NULDS))**2+(XB-XGLLD(NULDS))**2)
      P(NULDS)=P(NULDS)+FACTOR*((ENDLD +STALME(NULDS))/2.)
  300 CONTINUE
C     THERE ARE NULDS CONCENTRATED LOADS
      IOP=0
      DO 320 K=1,NULDS
      IF (KINDLO(K)-1) 320,301,301
  301 CONTINUE
C     JOINT LOADS
      IF (IOP-1) 3011,3012,3012
 3011 WRITE (7,10)
```

```
 3012 IOP=IOP+1
      GO TO (302,303,304),NLOTY
  302 NULOM9=NULOME(K)
      P9=P(K)
      WRITE (7,11) NULOM9,P9
      GO TO 320
  303 NULOM9=NULOME(K)
      P9=P(K)
      WRITE (7,12) NULOM9,P9
      GO TO 320
  304 NULOM9=NULOME(K)
      P9=P(K)
      WRITE (7,13) NULOM9,P9
  320 CONTINUE
C     JOINT LOADS O.K. MEMBER LOADS NOW
      IOP=0
      DO 340 K=1,NULDS
      IF (KINDLO(K)-1) 321,340,340
  321 CONTINUE
C     MEMBER CONCENTRATED LOADS
      IF (IOP-1) 3211,3212,3212
 3211 WRITE (7,14)
 3212 IOP=IOP+1
      GO TO (322,323,324),NLOTY
  322 NULOM9=NULOME(K)
      P9=P(K)
      XLOCA9=XLOCAL(K)
      WRITE (7,15) NULOM9,P9,XLOCA9
      GO TO 340
  323 NULOM9=NULOME(K)
      P9=P(K)
      XLOCA9=XLOCAL(K)
      WRITE (7,16) NULOM9,P9,XLOCA9
      GO TO 340
  324 NULOM9=NULOME(K)
      P9=P(K)
      XLOCA9=XLOCAL(K)
      WRITE (7,17) NULOM9,P9,XLOCA9
  340 CONTINUE
C     THIS LOADING ARRANGED.TRY NEXT
      GO TO 370
  350 CONTINUE
      GO TO (3501,3501,3501,2103,3501,3501,2103,3501),KIOUT
 3501 CONTINUE
C     MEMBER LINE LOADS
      NULD=NULDS+1
      STALME(NULD)=ENDLD
      IOP=0
      DO 360 L=1,NULDS
      I=L+1
      IF (IOP-1) 351,352,352
  351 WRITE (7,14)
  352 IOP=IOP+1
      GO TO (353,354,355),NLOTY
  353 CONTINUE
      IF (LTYPE-1) 3531,3532,3531
 3531 NULOM9=NULOME(L)
      STALM9=STALME(L)
      XLOCA9=XLOCAL(L)
```

```
      YLOCA9=YLOCAL(L)
      WRITE (7,18) NULOM9,STALM9,XLOCA9,YLOCA9
      GO TO 360
 3532 NULOM9=NULOME(L)
      STALM9=STALME(L)
      STALM8=STALME(I)
      XLOCA9=XLOCAL(L)
      YLOCA9=YLOCAL(L)
      WRITE (7,21) NULOM9,STALM9,STALM8,XLOCA9,YLOCA9
      GO TO 360
  354 CONTINUE
      IF (LTYPE-1) 3541,3542,3541
 3541 NULOM9=NULOME(L)
      STALM9=STALME(L)
      XLOCA9=XLOCAL(L)
      YLOCA9=YLOCAL(L)
      WRITE (7,19) NULOM9,STALM9,XLOCA9,YLOCA9
      GO TO 360
 3542 NULOM9=NULOME(L)
      STALM9=STALME(L)
      STALM8=STALME(I)
      XLOCA9=XLOCAL(L)
      YLOCA9=YLOCAL(L)
      WRITE (7,22) NULOM9,STALM9,STALM8,XLOCA9,YLOCA9
      GO TO 360
  355 CONTINUE
      IF (LTYPE-1) 3551,3552,3551
 3551 NULOM9=NULOME(L)
      STALM9=STALME(L)
      XLOCA9=XLOCAL(L)
      YLOCA9=YLOCAL(L)
      WRITE (7,20) NULOM9,STALM9,XLOCA9,YLOCA9
      GO TO 360
 3552 NULOM9=NULOME(L)
      STALM9=STALME(L)
      STALM8=STALME(I)
      XLOCA9=XLOCAL(L)
      YLOCA9=YLOCAL(L)
      WRITE (7,23) NULOM9,STALM9,STALM8,XLOCA9,YLOCA9
  360 CONTINUE
  370 CONTINUE
      DESTROY RADIUS
      DESTROY STALME
      DESTROY JOINT
      DESTROY P
      RETURN
      END
```

```
**026
      SUBROUTINE AREALO
      DYNAMIC ARRAY HOCOR,VECOR,NUJO,JORENU,KINDJO,NUPLJO,NUMIJO,NUMEMB
      DYNAMIC ARRAY KINDME,NUJO1,HOCOR1,VECOR1
      DYNAMIC ARRAY YMEMB,XSTAME,STALDM,ENDLDM
      COMMON HOLEN,VELEN,THICK,MANUJO,YOUNG,PRATIO,HOLEN1,HOLEN2,VELEN1
     1,VELEN2,EX,WHY,KIND1,EX2,WHY2,POLOM1,KILILO,LTYPE,NUHOJ1,NUVEJ1,
     2NJ,HOCOR(P),VECOR(P),NUJO(P),JORENU(P),HOLEN3,VELEN3,NUHOJ2,NUHOJ3
     3,NUVEJ2,NUVEJ3,NUPLJO(P),NUMIJO(P),NUMEMB(P),KINDME(P),NUME,NUVEMB
     4,NUHOMB,NURIDI,NULEDI,NUPOLO,NUJO1(P),HOCOR1(P),VECOR1(P),NSIZE,
     5NULDS,ENDLD,LDMSTA,LDMEND,KINDJO(P),KIOUT,NLOTY
      REAL LDMSTA,LDMEND
      EQUIVALENCE (EX,XA),(WHY,YA),(EX2,XB),(WHY2,YB),(STALD,POLOM1)
      DEFINE YMEMB,NUHOMB+NUVEMB
      DEFINE XSTAME,NUHOMB+NUVEMB
      DEFINE STALDM,NUHOMB+NUVEMB
      DEFINE ENDLDM,NUHOMB+NUVEMB
      IF (LTYPE-1) 10,10,20
   10 CONTINUE
C     LOADING UNIFORM OR CHANGING WITH X.DISTRIBUTE AMONG HORIZ.MEMBERS.
      K=0
      DO 100 INDEX=1,NUHOMB
      I=INDEX+NUVEMB
      NALPHA=NUMIJO(I)
      NBETA=NUPLJO(I)
      IF (XA-HOCOR(NBETA)) 11,100,100
   11 CONTINUE
      IF (XB-HOCOR(NALPHA)) 100,100,12
   12 CONTINUE
C     OVERLAP POSSIBLE BUT NOT CERTAIN
      IF (YA-VECOR(NBETA))13,13,100
   13 CONTINUE
C     OVERLAP STILL POSSIBLE
      IF (YB-VECOR(NALPHA)) 100,14,14
   14 CONTINUE
C     OVERLAP CERTAIN ,RECORD Y.
      K=K+1
      YMEMB(K)=VECOR(NBETA)
      XSTAME(K)=HOCOR(NALPHA)
  100 CONTINUE
      N=1
      DO 110 I=1,K
      L=I+1
      IF (XSTAME(L)-XSTAME(I)) 129,108,129
  108 N=N+1
  110 CONTINUE
  129 ADENO=YB-YA
      Y1=YA
      Y2=YB
  130 NUMBER=N
      IF (NUMBER-1) 131,132,131
  132 CONTINUE
      STALDM(1)=LDMSTA*ADENO
      ENDLDM(1)=LDMEND*ADENO
      GO TO 150
  131 CONTINUE
      DO 150 I=1,N
      L=I+1
      M=I-1
```

```
      IF (I-1) 134,133,134
  133 CONTINUE
      STALDM(I)=(((YMEMB(I)-Y1)+(YMEMB(L)-YMEMB(I))/2.)*LDMSTA)
      ENDLDM(I)=(((YMEMB(I)-Y1)+(YMEMB(L)-YMEMB(I))/2.)*LDMEND)
      GO TO 150
  134 CONTINUE
      IF (I-N) 135,137,135
  135 CONTINUE
      STALDM(I)=((YMEMB(L)-YMEMB(M))/2.)*LDMSTA
      ENDLDM(I)=((YMEMB(L)-YMEMB(M))/2.)*LDMEND
      GO TO 150
  137 CONTINUE
      STALDM(I)=((YMEMB(I)-YMEMB(M))/2.+(Y2-YMEMB(I)))*LDMSTA
      ENDLDM(I)=((YMEMB(I)-YMEMB(M))/2.+(Y2-YMEMB(I)))*LDMEND
  150 CONTINUE
C     LOADS READY TO CALL LINELO
      IF (LTYPE-1) 30,30,50
   30 CONTINUE
C     UNIFORM LOAD OR LINEAR ALONG X.
      DO 31 I=1,NUMBER
      YA=YMEMB(I)
      YB=YMEMB(I)
      STALD=STALDM(I)
      ENDLD=ENDLDM(I)
      CALL LINELO
   31 CONTINUE
      GO TO 300
   50 CONTINUE
C     LINEAR ALONG Y-AXIS.
      DO 51 I=1,NUMBER
      XA=YMEMB(I)
      XB=YMEMB(I)
      STALD=STALDM(I)
      ENDLD=ENDLDM(I)
      LTYPE=1
      CALL LINELO
   51 CONTINUE
      GO TO 300
   20 CONTINUE
C     LOADING LINEAR CHANGING WITH Y.DISTRIBUTE AMONG VERT.MEMBERS.
      K=0
      DO 200 I=1,NUVEMB
      NALPHA=NUMIJO(I)
      NBETA=NUPLJO(I)
      IF (YA   -VECOR(NBETA)) 21,200,200
   21 CONTINUE
      IF (YB   -VECOR(NALPHA))  200,200,22
   22 CONTINUE
C     OVERLAP POSSIBLE BUT NOT CERTAIN
      IF (XA   -HOCOR(NBETA)) 23,23,200
   23 CONTINUE
C     OVERLAP STILL POSSIBLE
      IF (XB   -HOCOR(NALPHA)) 200,24,24
   24 CONTINUE
C     OVERLAP CERTAIN RECORD X.
      K=K+1
      YMEMB(K)=HOCOR(NBETA)
      XSTAME(K)=VECOR(NALPHA)
  200 CONTINUE
```

```
      N=0
      DO 210 I=1,K
      IF (XSTAME(I)-XSTAME(1)) 210,208,210
  208 N=N+1
      XSTAME(N)=XSTAME(I)
      YMEMB(N)=YMEMB(I)
  210 CONTINUE
  229 ADENO=XB-XA
      Y1=XA
      Y2=XB
      GO TO 130
  300 CONTINUE
      DESTROY YMEMB
      DESTROY XSTAME
      DESTROY STALDM
      DESTROY ENDLDM
      RETURN
      END
```

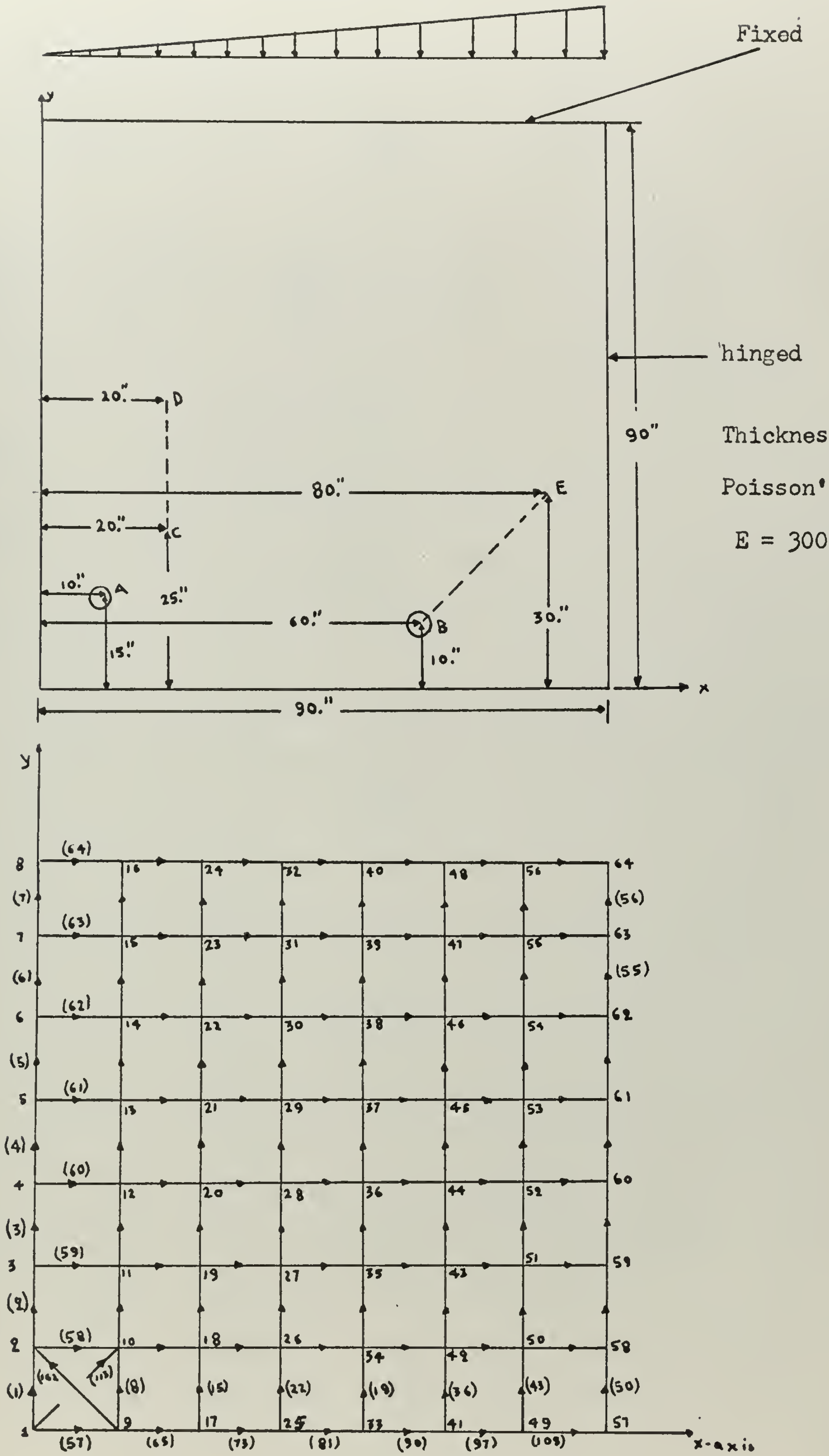
```
**026
      SUBROUTINE OUT
      DYNAMIC ARRAY HOCOR,VECOR,NUJO,JORENO,KINDJO,NUPLJO,NUMIJO,NUMEMB
      DYNAMIC ARRAY KINDME,NUJO1,HOCOR1,VECOR1
      COMMON HOLEN,VELEN,THICK,MANUJO,YOUNG,PRATIO,HOLEN1,HOLEN2,VELEN1
     1,VELEN2,EX,WHY,KIND1,EX2,WHY2,POLOM1,KILILO,LTYPE,NOHOJ1,NOVEJ1,
     2NJ,HOCOR(P),VECOR(P),NUJO(P),JORENO(P),HOLEN3,VELEN3,NOHOJ2,NOHOJ3
     3,NOVEJ2,NOVEJ3,NUPLJO(P),NUMIJO(P),NUMEMB(P),KINDME(P),NUME,NOVEMB
     4,NOHOMB,NORIDI,NOLELI,NUPOLO,NUJO1(P),HOCOR1(P),VECOR1(P),NSIZE,
     5NOLDS,ENDLD,LDMSTA,LDMEND,KINDJO(P),KIOUT,NLOTY
    1 FORMAT (T8,15H LIST REACTIONS)
    2 FORMAT (T8,11H LIST LOADS)
    3 FORMAT (T8,19H LIST DISPLACEMENTS)
    4 FORMAT (T8,20H STIFFNESS ANALYSIS )
    6 FORMAT (T8,17H LOADING LIST '1')
      IF (MANUJO) 10,5,10
    5 CONTINUE
      WRITE (7,6)
      WRITE (7,4)
      MANUJO=MANUJO+1
   10 CONTINUE
      GO TO (20,20,30,40,50,30,40,80),KIOUT
   20 CONTINUE
C     REACTIONS WANTED
      WRITE (7,1)
      GO TO 1000
   30 CONTINUE
C     POINT LOADS OR DISPLACEMENTS WANTED
      NUPOLO=2
      LOADMA=0.
      CALL POINLO
      GO TO 1000
   40 CONTINUE
C     LINE LOADS WANTED
      NLOTY=1
      CALL LINELO
      GO TO 1000
   50 CONTINUE
C     ALL LOADS WANTED
      WRITE (7,2)
      GO TO 1000
   80 CONTINUE
C     ALL DISPLACEMENTS WANTED
      WRITE (7,3)
 1000 CONTINUE
      RETURN
      END
```

APPENDIX C

INPUT AND OUTPUT OF SAMPLE PROBLEM

Fig. C.1

SAMPLE PROBLEM



Thickness = 0.5

Poisson's Ratio = 0.3

E = 30000

SAMPLE PROBLEM-DATA

```
/DATA
    90.        90.          .5    3000          .3
    65
   1
       4
   1    0.0       0.0       0.0       90.0
   1   90.        0.0      90.        90.
   2    0.0       0.0      90.0        0.0
   2    0.0      90.       90.        90.0
           1
    1     0.0       0.0       0.0      1    90.       90.        -10.0
   7
    2
    3   10.       15.
    4   20.       25.       20.       45.
    5
    6   60.       10.
    7   60.       10.       80.       30.
    8
**EOF
/END OF FILE
```

```
 SAMPLE PROBLEM.

 SQUARE PLATE 90.X90. INCHES.
 THICKNESS=0.5 INCHES.
 TWO CLAMPED,TWO HINGED EDGES.
 YOUNG'S MODULUS=30000. KIPS
 POISSON'S RATIO=0.3
 HYDROSTATIC LOADING.



 TYPE PLANE GRID XY
 CONSTANTS E ,    30000.000,ALL
 CONSTANTS G ,    11538.465,ALL
 UNITS,INCHES,KIPS
JOINT COORDINATES
   1     0.0         0.0     SUPPORT
   2     0.0        12.8571  SUPPORT
   3     0.0        25.7143  SUPPORT
   4     0.0        38.5714  SUPPORT
   5     0.0        51.4286  SUPPORT
   6     0.0        64.2857  SUPPORT
   7     0.0        77.1429  SUPPORT
   8     0.0        90.0000  SUPPORT
   9    12.8571      0.0     SUPPORT
  10    12.8571     12.8571
  11    12.8571     25.7143
  12    12.8571     38.5714
  13    12.8571     51.4286
  14    12.8571     64.2857
  15    12.8571     77.1429
  16    12.8571     90.0000  SUPPORT
  17    25.7143      0.0     SUPPORT
  18    25.7143     12.8571
  19    25.7143     25.7143
  20    25.7143     38.5714
  21    25.7143     51.4286
  22    25.7143     64.2857
  23    25.7143     77.1429
  24    25.7143     90.0000  SUPPORT
  25    38.5714      0.0     SUPPORT
  26    38.5714     12.8571
  27    38.5714     25.7143
  28    38.5714     38.5714
  29    38.5714     51.4286
  30    38.5714     64.2857
  31    38.5714     77.1429
  32    38.5714     90.0000  SUPPORT
  33    51.4286      0.0     SUPPORT
  34    51.4286     12.8571
  35    51.4286     25.7143
  36    51.4286     38.5714
  37    51.4286     51.4286
  38    51.4286     64.2857
  39    51.4286     77.1429
```

```
   40    51.4286    90.0000 SUPPORT
   41    64.2857     0.0    SUPPORT
   42    64.2857    12.8571
   43    64.2857    25.7143
   44    64.2857    38.5714
   45    64.2857    51.4286
   46    64.2857    64.2857
   47    64.2857    77.1429
   48    64.2857    90.0000 SUPPORT
   49    77.1429     0.0    SUPPORT
   50    77.1429    12.8571
   51    77.1429    25.7143
   52    77.1429    38.5714
   53    77.1429    51.4286
   54    77.1429    64.2857
   55    77.1429    77.1429
   56    77.1429    90.0000 SUPPORT
   57    90.0000     0.0    SUPPORT
   58    90.0000    12.8571 SUPPORT
   59    90.0000    25.7143 SUPPORT
   60    90.0000    38.5714 SUPPORT
   61    90.0000    51.4286 SUPPORT
   62    90.0000    64.2857 SUPPORT
   63    90.0000    77.1429 SUPPORT
   64    90.0000    90.0000 SUPPORT
JOINT RELEASES
    2 MOMENT Y
    3 MOMENT Y
    4 MOMENT Y
    5 MOMENT Y
    6 MOMENT Y
    7 MOMENT Y
   58 MOMENT Y
   59 MOMENT Y
   60 MOMENT Y
   61 MOMENT Y
   62 MOMENT Y
   63 MOMENT Y
 MEMBER INCIDENCES
    1    1    2
    2    2    3
    3    3    4
    4    4    5
    5    5    6
    6    6    7
    7    7    8
    8    9   10
    9   10   11
   10   11   12
   11   12   13
   12   13   14
   13   14   15
   14   15   16
   15   17   18
   16   18   19
```

```
17  19  20
18  20  21
19  21  22
20  22  23
21  23  24
22  25  26
23  26  27
24  27  28
25  28  29
26  29  30
27  30  31
28  31  32
29  33  34
30  34  35
31  35  36
32  36  37
33  37  38
34  38  39
35  39  40
36  41  42
37  42  43
38  43  44
39  44  45
40  45  46
41  46  47
42  47  48
43  49  50
44  50  51
45  51  52
46  52  53
47  53  54
48  54  55
49  55  56
50  57  58
51  58  59
52  59  60
53  60  61
54  61  62
55  62  63
56  63  64
57   1   9
58   2  10
59   3  11
60   4  12
61   5  13
62   6  14
63   7  15
64   8  16
65   9  17
66  10  18
67  11  19
68  12  20
69  13  21
70  14  22
71  15  23
```

```
 72  16  24
 73  17  25
 74  18  26
 75  19  27
 76  20  28
 77  21  29
 78  22  30
 79  23  31
 80  24  32
 81  25  33
 82  26  34
 83  27  35
 84  28  36
 85  29  37
 86  30  38
 87  31  39
 88  32  40
 89  33  41
 90  34  42
 91  35  43
 92  36  44
 93  37  45
 94  38  46
 95  39  47
 96  40  48
 97  41  49
 98  42  50
 99  43  51
100  44  52
101  45  53
102  46  54
103  47  55
104  48  56
105  49  57
106  50  58
107  51  59
108  52  60
109  53  61
110  54  62
111  55  63
112  56  64
113   1  10
114   2  11
115   3  12
116   4  13
117   5  14
118   6  15
119   7  16
120   9  18
121  10  19
122  11  20
123  12  21
124  13  22
125  14  23
126  15  24
```

127	17	26
128	18	27
129	19	28
130	20	29
131	21	30
132	22	31
133	23	32
134	25	34
135	26	35
136	27	36
137	28	37
138	29	38
139	30	39
140	31	40
141	33	42
142	34	43
143	35	44
144	36	45
145	37	46
146	38	47
147	39	48
148	41	50
149	42	51
150	43	52
151	44	53
152	45	54
153	46	55
154	47	56
155	49	58
156	50	59
157	51	60
158	52	61
159	53	62
160	54	63
161	55	64
162	9	2
163	10	3
164	11	4
165	12	5
166	13	6
167	14	7
168	15	8
169	17	10
170	18	11
171	19	12
172	20	13
173	21	14
174	22	15
175	23	16
176	25	18
177	26	19
178	27	20
179	28	21
180	29	22
181	30	23

```
 182  31  24
 183  33  26
 184  34  27
 185  35  28
 186  36  29
 187  37  30
 188  38  31
 189  39  32
 190  41  34
 191  42  35
 192  43  36
 193  44  37
 194  45  38
 195  46  39
 196  47  40
 197  49  42
 198  50  43
 199  51  44
 200  52  45
 201  53  46
 202  54  47
 203  55  48
 204  57  50
 205  58  51
 206  59  52
 207  60  53
 208  61  54
 209  62  55
 210  63  56
MEMBER PROPERTIES,PRISMATIC
    1 IY         0.05151 IX        0.01913
    2 IY         0.05151 IX        0.01913
    3 IY         0.05151 IX        0.01913
    4 IY         0.05151 IX        0.01913
    5 IY         0.05151 IX        0.01913
    6 IY         0.05151 IX        0.01913
    7 IY         0.05151 IX        0.01913
    8 IY         0.10302 IX        0.03827
    9 IY         0.10302 IX        0.03827
   10 IY         0.10302 IX        0.03827
   11 IY         0.10302 IX        0.03827
   12 IY         0.10302 IX        0.03827
   13 IY         0.10302 IX        0.03827
   14 IY         0.10302 IX        0.03827
   15 IY         0.10302 IX        0.03827
   16 IY         0.10302 IX        0.03827
   17 IY         0.10302 IX        0.03827
   18 IY         0.10302 IX        0.03827
   19 IY         0.10302 IX        0.03827
   20 IY         0.10302 IX        0.03827
   21 IY         0.10302 IX        0.03827
   22 IY         0.10302 IX        0.03827
   23 IY         0.10302 IX        0.03827
   24 IY         0.10302 IX        0.03827
   25 IY         0.10302 IX        0.03827
```

```
26 IY          0.10302 IX          0.03827
27 IY          0.10302 IX          0.03827
28 IY          0.10302 IX          0.03827
29 IY          0.10302 IX          0.03827
30 IY          0.10302 IX          0.03827
31 IY          0.10302 IX          0.03827
32 IY          0.10302 IX          0.03827
33 IY          0.10302 IX          0.03827
34 IY          0.10302 IX          0.03827
35 IY          0.10302 IX          0.03827
36 IY          0.10302 IX          0.03827
37 IY          0.10302 IX          0.03827
38 IY          0.10302 IX          0.03827
39 IY          0.10302 IX          0.03827
40 IY          0.10302 IX          0.03827
41 IY          0.10302 IX          0.03827
42 IY          0.10302 IX          0.03827
43 IY          0.10302 IX          0.03827
44 IY          0.10302 IX          0.03827
45 IY          0.10302 IX          0.03827
46 IY          0.10302 IX          0.03827
47 IY          0.10302 IX          0.03827
48 IY          0.10302 IX          0.03827
49 IY          0.10302 IX          0.03827
50 IY          0.05151 IX          0.01913
51 IY          0.05151 IX          0.01913
52 IY          0.05151 IX          0.01913
53 IY          0.05151 IX          0.01913
54 IY          0.05151 IX          0.01913
55 IY          0.05151 IX          0.01913
56 IY          0.05151 IX          0.01913
57 IY          0.05151 IX          0.01913
58 IY          0.10302 IX          0.03827
59 IY          0.10302 IX          0.03827
60 IY          0.10302 IX          0.03827
61 IY          0.10302 IX          0.03827
62 IY          0.10302 IX          0.03827
63 IY          0.10302 IX          0.03827
64 IY          0.05151 IX          0.01913
65 IY          0.05151 IX          0.01913
66 IY          0.10302 IX          0.03827
67 IY          0.10302 IX          0.03827
68 IY          0.10302 IX          0.03827
69 IY          0.10302 IX          0.03827
70 IY          0.10302 IX          0.03827
71 IY          0.10302 IX          0.03827
72 IY          0.05151 IX          0.01913
73 IY          0.05151 IX          0.01913
74 IY          0.10302 IX          0.03827
75 IY          0.10302 IX          0.03827
76 IY          0.10302 IX          0.03827
77 IY          0.10302 IX          0.03827
78 IY          0.10302 IX          0.03827
79 IY          0.10302 IX          0.03827
80 IY          0.05151 IX          0.01913
```

```
 81 IY          0.05151 IX          0.01913
 82 IY          0.10302 IX          0.03827
 83 IY          0.10302 IX          0.03827
 84 IY          0.10302 IX          0.03827
 85 IY          0.10302 IX          0.03827
 86 IY          0.10302 IX          0.03827
 87 IY          0.10302 IX          0.03827
 88 IY          0.05151 IX          0.01913
 89 IY          0.05151 IX          0.01913
 90 IY          0.10302 IX          0.03827
 91 IY          0.10302 IX          0.03827
 92 IY          0.10302 IX          0.03827
 93 IY          0.10302 IX          0.03827
 94 IY          0.10302 IX          0.03827
 95 IY          0.10302 IX          0.03827
 96 IY          0.05151 IX          0.01913
 97 IY          0.05151 IX          0.01913
 98 IY          0.10302 IX          0.03827
 99 IY          0.10302 IX          0.03827
100 IY          0.10302 IX          0.03827
101 IY          0.10302 IX          0.03827
102 IY          0.10302 IX          0.03827
103 IY          0.10302 IX          0.03827
104 IY          0.05151 IX          0.01913
105 IY          0.05151 IX          0.01913
106 IY          0.10302 IX          0.03827
107 IY          0.10302 IX          0.03827
108 IY          0.10302 IX          0.03827
109 IY          0.10302 IX          0.03827
110 IY          0.10302 IX          0.03827
111 IY          0.10302 IX          0.03827
112 IY          0.05151 IX          0.01913
113 IY          0.06244 IX          0.0
114 IY          0.06244 IX          0.0
115 IY          0.06244 IX          0.0
116 IY          0.06244 IX          0.0
117 IY          0.06244 IX          0.0
118 IY          0.06244 IX          0.0
119 IY          0.06244 IX          0.0
120 IY          0.06244 IX          0.0
121 IY          0.06244 IX          0.0
122 IY          0.06244 IX          0.0
123 IY          0.06244 IX          0.0
124 IY          0.06244 IX          0.0
125 IY          0.06244 IX          0.0
126 IY          0.06244 IX          0.0
127 IY          0.06244 IX          0.0
128 IY          0.06244 IX          0.0
129 IY          0.06244 IX          0.0
130 IY          0.06244 IX          0.0
131 IY          0.06244 IX          0.0
132 IY          0.06244 IX          0.0
133 IY          0.06244 IX          0.0
134 IY          0.06244 IX          0.0
135 IY          0.06244 IX          0.0
```

```
136 IY          0.06244  IX        0.0
137 IY          0.06244  IX        0.0
138 IY          0.06244  IX        0.0
139 IY          0.06244  IX        0.0
140 IY          0.06244  IX        0.0
141 IY          0.06244  IX        0.0
142 IY          0.06244  IX        0.0
143 IY          0.06244  IX        0.0
144 IY          0.06244  IX        0.0
145 IY          0.06244  IX        0.0
146 IY          0.06244  IX        0.0
147 IY          0.06244  IX        0.0
148 IY          0.06244  IX        0.0
149 IY          0.06244  IX        0.0
150 IY          0.06244  IX        0.0
151 IY          0.06244  IX        0.0
152 IY          0.06244  IX        0.0
153 IY          0.06244  IX        0.0
154 IY          0.06244  IX        0.0
155 IY          0.06244  IX        0.0
156 IY          0.06244  IX        0.0
157 IY          0.06244  IX        0.0
158 IY          0.06244  IX        0.0
159 IY          0.06244  IX        0.0
160 IY          0.06244  IX        0.0
161 IY          0.06244  IX        0.0
162 IY          0.06244  IX        0.0
163 IY          0.06244  IX        0.0
164 IY          0.06244  IX        0.0
165 IY          0.06244  IX        0.0
166 IY          0.06244  IX        0.0
167 IY          0.06244  IX        0.0
168 IY          0.06244  IX        0.0
169 IY          0.06244  IX        0.0
170 IY          0.06244  IX        0.0
171 IY          0.06244  IX        0.0
172 IY          0.06244  IX        0.0
173 IY          0.06244  IX        0.0
174 IY          0.06244  IX        0.0
175 IY          0.06244  IX        0.0
176 IY          0.06244  IX        0.0
177 IY          0.06244  IX        0.0
178 IY          0.06244  IX        0.0
179 IY          0.06244  IX        0.0
180 IY          0.06244  IX        0.0
181 IY          0.06244  IX        0.0
182 IY          0.06244  IX        0.0
183 IY          0.06244  IX        0.0
184 IY          0.06244  IX        0.0
185 IY          0.06244  IX        0.0
186 IY          0.06244  IX        0.0
187 IY          0.06244  IX        0.0
188 IY          0.06244  IX        0.0
189 IY          0.06244  IX        0.0
190 IY          0.06244  IX        0.0
```

```
191 IY          0.06244 IX          0.0
192 IY          0.06244 IX          0.0
193 IY          0.06244 IX          0.0
194 IY          0.06244 IX          0.0
195 IY          0.06244 IX          0.0
196 IY          0.06244 IX          0.0
197 IY          0.06244 IX          0.0
198 IY          0.06244 IX          0.0
199 IY          0.06244 IX          0.0
200 IY          0.06244 IX          0.0
201 IY          0.06244 IX          0.0
202 IY          0.06244 IX          0.0
203 IY          0.06244 IX          0.0
204 IY          0.06244 IX          0.0
205 IY          0.06244 IX          0.0
206 IY          0.06244 IX          0.0
207 IY          0.06244 IX          0.0
208 IY          0.06244 IX          0.0
209 IY          0.06244 IX          0.0
210 IY          0.06244 IX          0.0
LOADING '1'
MEMBER LOADS
 57 FORCE Z LINEAR WA          0.0    WB          -9.184 -
LA          0.0    LB         12.857
 65 FORCE Z LINEAR WA         -9.184  WB         -18.367 -
LA          0.0    LB         12.857
 73 FORCE Z LINEAR WA        -18.367  WB         -27.551 -
LA          0.0    LB         12.857
 81 FORCE Z LINEAR WA        -27.551  WB         -36.735 -
LA          0.0    LB         12.857
 89 FORCE Z LINEAR WA        -36.735  WB         -45.918 -
LA          0.0    LB         12.857
 97 FORCE Z LINEAR WA        -45.918  WB         -55.102 -
LA          0.0    LB         12.857
105 FORCE Z LINEAR WA        -55.102  WB         -64.286 -
LA          0.0    LB         12.857
MEMBER LOADS
 58 FORCE Z LINEAR WA          0.0    WB         -18.367 -
LA          0.0    LB         12.857
 66 FORCE Z LINEAR WA        -18.367  WB         -36.735 -
LA          0.0    LB         12.857
 74 FORCE Z LINEAR WA        -36.735  WB         -55.102 -
LA          0.0    LB         12.857
 82 FORCE Z LINEAR WA        -55.102  WB         -73.469 -
LA          0.0    LB         12.857
 90 FORCE Z LINEAR WA        -73.469  WB         -91.837 -
LA          0.0    LB         12.857
 98 FORCE Z LINEAR WA        -91.837  WB        -110.204 -
LA          0.0    LB         12.857
106 FORCE Z LINEAR WA       -110.204  WB        -128.571 -
LA          0.0    LB         12.857
MEMBER LOADS
 59 FORCE Z LINEAR WA          0.0    WB         -18.367 -
LA          0.0    LB         12.857
 67 FORCE Z LINEAR WA        -18.367  WB         -36.735 -
```

```
LA           0.0   LB           12.857
 75 FORCE Z LINEAR WA          -36.735 WB          -55.102 -
LA           0.0   LB           12.857
 83 FORCE Z LINEAR WA          -55.102 WB          -73.469 -
LA           0.0   LB           12.857
 91 FORCE Z LINEAR WA          -73.469 WB          -91.837 -
LA           0.0   LB           12.857
 99 FORCE Z LINEAR WA          -91.837 WB         -110.204 -
LA           0.0   LB           12.857
107 FORCE Z LINEAR WA         -110.204 WB         -128.571 -
LA           0.0   LB           12.857
MEMBER LOADS
 60 FORCE Z LINEAR WA            0.0    WB          -18.367 -
LA           0.0   LB           12.857
 68 FORCE Z LINEAR WA          -18.367 WB          -36.735 -
LA           0.0   LB           12.857
 76 FORCE Z LINEAR WA          -36.735 WB          -55.102 -
LA           0.0   LB           12.857
 84 FORCE Z LINEAR WA          -55.102 WB          -73.469 -
LA           0.0   LB           12.857
 92 FORCE Z LINEAR WA          -73.469 WB          -91.837 -
LA           0.0   LB           12.857
100 FORCE Z LINEAR WA          -91.837 WB         -110.204 -
LA           0.0   LB           12.857
108 FORCE Z LINEAR WA         -110.204 WB         -128.571 -
LA           0.0   LB           12.857
MEMBER LOADS
 61 FORCE Z LINEAR WA            0.0    WB          -18.367 -
LA           0.0   LB           12.857
 69 FORCE Z LINEAR WA          -18.367 WB          -36.735 -
LA           0.0   LB           12.857
 77 FORCE Z LINEAR WA          -36.735 WB          -55.102 -
LA           0.0   LB           12.857
 85 FORCE Z LINEAR WA          -55.102 WB          -73.469 -
LA           0.0   LB           12.857
 93 FORCE Z LINEAR WA          -73.469 WB          -91.837 -
LA           0.0   LB           12.857
101 FORCE Z LINEAR WA          -91.837 WB         -110.204 -
LA           0.0   LB           12.857
109 FORCE Z LINEAR WA         -110.204 WB         -128.571 -
LA           0.0   LB           12.857
MEMBER LOADS
 62 FORCE Z LINEAR WA            0.0    WB          -18.367 -
LA           0.0   LB           12.857
 70 FORCE Z LINEAR WA          -18.367 WB          -36.735 -
LA           0.0   LB           12.857
 78 FORCE Z LINEAR WA          -36.735 WB          -55.102 -
LA           0.0   LB           12.857
 86 FORCE Z LINEAR WA          -55.102 WB          -73.469 -
LA           0.0   LB           12.857
 94 FORCE Z LINEAR WA          -73.469 WB          -91.837 -
LA           0.0   LB           12.857
102 FORCE Z LINEAR WA          -91.837 WB         -110.204 -
LA           0.0   LB           12.857
110 FORCE Z LINEAR WA         -110.204 WB         -128.571 -
```

```
LA           0.0   LB          12.857
MEMBER LOADS
 63 FORCE Z LINEAR WA           0.0   WB         -18.367 -
LA           0.0   LB          12.857
 71 FORCE Z LINEAR WA         -18.367 WB         -36.735 -
LA           0.0   LB          12.857
 79 FORCE Z LINEAR WA         -36.735 WB         -55.102 -
LA           0.0   LB          12.857
 87 FORCE Z LINEAR WA         -55.102 WB         -73.469 -
LA           0.0   LB          12.857
 95 FORCE Z LINEAR WA         -73.469 WB         -91.837 -
LA           0.0   LB          12.857
103 FORCE Z LINEAR WA         -91.837 WB        -110.204 -
LA           0.0   LB          12.857
111 FORCE Z LINEAR WA        -110.204 WB        -128.571 -
LA           0.0   LB          12.857
MEMBER LOADS
 64 FORCE Z LINEAR WA           0.0   WB          -9.184 -
LA           0.0   LB          12.857
 72 FORCE Z LINEAR WA          -9.184 WB         -18.367 -
LA           0.0   LB          12.857
 80 FORCE Z LINEAR WA         -18.367 WB         -27.551 -
LA           0.0   LB          12.857
 88 FORCE Z LINEAR WA         -27.551 WB         -36.735 -
LA           0.0   LB          12.857
 96 FORCE Z LINEAR WA         -36.735 WB         -45.918 -
LA           0.0   LB          12.857
104 FORCE Z LINEAR WA         -45.918 WB         -55.102 -
LA           0.0   LB          12.857
112 FORCE Z LINEAR WA         -55.102 WB         -64.286 -
LA           0.0   LB          12.857
LOADING LIST '1'
STIFFNESS ANALYSIS
LIST REACTIONS
LIST LOADS  10
LIST LOADS  19
LIST LOADS  20
LIST LOADS
LIST DISPLACEMENTS   42
LIST DISPLACEMENTS   42
LIST DISPLACEMENTS   51
LIST DISPLACEMENTS
FINISH
```

APPENDIX D

CDL PROPOSED LISTING

```
/JOB          1673  KOLLINIATIS
/ASSIGN  3=ICES
/RESTORE CDP1
CDP
INITIATE 'PLATES' 'SOLUTION'
ADD 'SPAN' REAL 'HOLEN' REQUIRED
FILE
ADD 'WIDTH' REAL 'VELEN' REQUIRED
FILE
ADD 'THICK' REAL 'THICK' REQUIRED
FILE
ADD 'YOUNG' REAL 'YOUNG' STANDARD 30000.
FILE
ADD 'POISSON' REAL 'PRATIO' STANDARD .3
FILE
ADD 'MAX' INTEGER 'MANUJO' STANDARD 300
FILE
ADD 'HOLE
ID 'AT X' REAL 'HOLEN1' REQUIRED
ID 'AND Y' REAL 'VELEN1' REQUIRED
ID 'SPAN' REAL 'HOLEN2' REQUIRED
ID 'WIDTH' REAL 'VELEN2' REQUIRED
EXECUTE 'DIPOHO' 'MUGRID' 'MEINHO'
FILE
ADD 'POINT S'
REPEAT TABULAR
ID 'X' REAL 'EX' REQUIRED
ID 'Y' REAL 'WHY' REQUIRED
```

```
EXISTENCE 'FREE' 'HINGED' 'FIXED' SET 'KIND1' STANDARD 0
EXECUTE 'POBOC'
END REPEAT
FILE
ADD 'LINE S'
REPEAT TABULAR
ID 'X1' REAL 'EX' REQUIRED
ID 'Y1' REAL 'WHY' REQUIRED
ID 'X2' REAL 'EX2' REQUIRED
ID 'Y2' REAL 'WHY2' REQUIRED
EXISTENCE 'FREE' 'HINGED' 'FIXED' SET 'KIND1' STANDARD 0
EXECUTE 'LIBOC'
END REPEAT
FILE
ADD 'FORM G'
EXECUTE 'JOINRE' 'MEMPRO'
FILE
ADD 'POINT L'
PRESET INTEGER 'KIOUT' EQUAL 1
PRESET INTEGER 'NUPOLO' EQUAL 1
REPEAT TABULAR
EXISTENCE 'NOTHING' 'Z FORCE' 'X MOMENT' 'Y MOMENT' 'SET 'NLOTY' -
REQUIRED
ID 'X' REAL 'EX' REQUIRED
ID 'Y' REAL 'WHY' REQUIRED
ID 'P' REAL 'STALD' REQUIRED
EXECUTE 'POINLO'
END REPEAT
FILE
ADD 'LINE L'
```

```
PRESET INTEGER 'KIOUT'EQUAL 1
REPEAT TABULAR
EXISTENCE 'FORCE Z LINEAR' 'FORCE Z UNIFORM' 'MOMENT X LINEAR'-
'MOMENT X UNIFORM' 'MOMENT Y LINEAR' 'MOMENT Y LINEAR'
'MOMENT X UNIFORM' 'MOMENT Y LINEAR' 'MOMENT Y UNIFORM'SET 'KILILO'-
REQUIRED
ID 'AT X' REAL 'EX' REQUIRED
ID 'AND Y' REAL 'WHY' REQUIRED
ID 'UP TO X' REAL 'EX2' REQUIRED
ID 'AND UP TO Y' REAL 'WHY2' REQUIRED
CONDITION 'KILILO' INTEGER EQUAL 0
PRESET INTEGER 'NLOTY' EQUAL 1
PRESET INTEGER 'LTYPE'EQUAL 1
EXECUTE 'LINELO'
OR CONDITION 'KILILO' INTEGER EQUAL 1
PRESET INTEGER 'NLOTY' EQUAL 1
PRESET INTEGER 'LTYPE'EQUAL 0
EXECUTE 'LINELO'
OR CONDITION 'KILILO' INTEGER EQUAL 2
PRESET INTEGER 'NLOTY' EQUAL 2
PRESET INTEGER 'LTYPE'EQUAL 1
EXECUTE 'LINELO'
OR CONDITION 'KILILO' INTEGER EQUAL 3
PRESET INTEGER 'NLOTY' EQUAL 2
PRESET INTEGER 'LTYPE'EQUAL 0
EXECUTE 'LINELO'
OR CONDITION 'KILILO' INTEGER EQUAL 4
PRESET INTEGER 'NLOTY' EQUAL 3
PRESET INTEGER 'LTYPE'EQUAL 1
EXECUTE 'LINELO'
```

```
OTHERWISE
PRESET INTEGER 'NLOTY' EQUAL 3
PRESET INTEGER 'LTYPE' EQUAL 1
EXECUTE LINELO
END REPEAT
FILE
ADD 'AREA L'
PRESET INTEGER 'KIOUT' EQUAL 1
REPEAT TABULAR
EXISTENCE 'FORCE Z UNIFORM' 'FORCE Z LINEAR ON X' 'FORCE Z LINEAR ON Y'-
'MOMENT X UNIFORM' 'MOMENT X LINEAR ON X' 'MOMENT X LINEAR ON Y'-
'MOMENT Y UNIFORM' 'MOMENT Y LINEAR ON X' 'MOMENT Y LINEAR ON Y'-
SET 'KILILO' REQUIRED
ID 'XA' REAL 'EX' REQUIRED
ID 'YA' REAL 'WHY' REQUIRED
ID 'WA' REAL 'LDMSTA' REQUIRED
ID 'XB' REAL 'EX2' REQUIRED
ID 'YB' REAL 'WHY2' REQUIRED
ID 'WB' REAL 'LDMEND' REQUIRED
CONTITION 'KILILO' INTEGER EQUAL 0
PRESET INTEGER 'NLOTY' EQUAL 1
PRESET INTEGER 'LTYPE'EQUAL 0
EXECUTE 'AREALO'
OR CONDITION 'KILILO' INTEGER EQUAL 1
PRESET INTEGER 'NLOTY' EQUAL 1
PRESET INEGER 'LTYPE' EQUAL 1
EXECUTE 'AREALO'
OR CONDITION 'KILILO' INTEGER EQUAL 2
PRESET INTEGER 'NLOTY' EQUAL 1
PRESET INTEGER 'LTYPE' EQUAL 2
```

```
EXECUTE 'AREALO'
OR CONDITION 'KILILO' INTEGER EQUAL 3
PRESET INTEGER 'NLOTY' EQUAL 2
PRESET INTEGER 'LTYPE' EQUAL 0
EXECUTE 'AREALO'
OR CONDITION 'KILILO' INTEGER EQUAL 4
PRESET INTEGER 'NLOTY' EQUAL 2
PRESET INTEGER 'LTYPE' EQUAL 1
EXECUTE 'AREALO'
OR CONDITION 'KILILO' INTEGER EQUAL 5
PRESET INTEGER 'NLOTY' EQUAL 2
PRESET INTEGER 'LTYPE' EQUAL 2
EXECUTE 'AREALO'
OR CONDITION 'KILILO' INTEGER EQUAL 6
PRESET INTEGER 'NLOTY' EQUAL 3
PRESET INTEGER 'LTYPE' EQUAL 0
EXECUTE 'AREALO'
OR CONDITION 'KILILO' INTEGER EQUAL 7
PRESET INTEGER 'NLOTY' EQUAL 3
PRESET INTEGER 'LTYPE' EQUAL 1
EXECUTE 'AREALO'
OR CONDITION 'KILILO' INTEGER EQUAL 8
PRESET INTEGER 'NLOTY' EQUAL 3
PRESET INTEGER 'LTYPE' EQUAL 2
EXECUTE 'AREALO'
END REPEAT
FILE
ADD 'LIST'
REPEAT TABULAR
MODIFIER 'REACTIONS'
```

```
PRESET INTEGER 'KIOUT' EQUAL 2
EXECUTE 'OUT'
OR MODIFIER 'POINT LOADS'
ID 'XA' REAL 'EX' REQUIRED
ID 'YA' REAL 'WHY' REQUIRED
PRESET INTEGER 'KIOUT' EQUAL 3
EXECUTE 'OUT'
OR MODIFIER 'LINE LOADS'
ID 'XA' REAL 'EX' REQUIRED
ID 'YA' REAL 'WHY' REQUIRED
ID 'XB' REAL 'EX2' REQUIRED
ID 'YB' REAL 'WHY2' REQUIRED
PRESET INTEGER 'KIOUT' EQUAL 4
EXECUTE 'OUT'
OR MODIFIER 'LOADS'
PRESET INTEGER 'KIOUT' EQUAL 5
EXECUTE 'OUT'
OR MODIFIER 'POINT DISPLACEMENTS'
ID 'XA' REAL 'EX' REQUIRED
ID 'YA' REAL 'WHY' REQUIRED
PRESET INTEGER 'KIOUT' EQUAL 6
EXECUTE 'OUT'
OR MODIFIER 'LINE DISPLACEMENTS'
ID 'XA' REAL 'EX' REQUIRED
ID 'YA' REAL 'WHY' REQUIRED
ID 'XB' REAL 'EX2' REQUIRED
ID 'YB' REAL 'WHY2' REQUIRED
PRESET INTEGER 'KIOUT' EQUAL 7
EXECUTE 'OUT'
OR MODIFIER 'DISPLACEMENTS'
```

```
PRESET INTEGER 'KIOUT' EQUAL 8
EXECUTE 'OUT'
END MODIFIER REQUIRED
END REPEAT
FILE
FINISH
/END OF FILE
```

APPENDIX E

A DICTIONARY OF TERMS IN COMMON

ENDLD = End load magnitude of line loads
EX = x-coordinate of point supports
EX2 = End x-coordinate of line supports
HOCOR = x-coordinate of joint
HOCOR1 = x-coordinate of a non support joint
HOLEN = Span of plating (parallel to the x-axis)
HOLEN1 = Distance along the x-axis from the center of coordinates up to the left edge of hole
HOLEN2 = Span of hole (parallel to the x-axis)
HOLEN3 = HOLEN - HOLEN1 - HOLEN2
JORENU = Joint release number
KINDJO = Kind of joint
KINDME = Kind of member
KIOUT = A number that specifies necessary output
KIND1 = Kind of point support
LDMSTA = Start load magnitude for area loads
LDMEND = End load magnitude for area loads
LTYPE = A number that specifies kind of load
MANUJO = Maximum number of joints
NJ = Number of joints
NUJO = Identification number of joint

NUJO1 = Identification number of non-support joint

NULDS = Number of grid loads used to approximate a given LINE - LOAD

NUMEMB = Identification number of member

NUMLJO = Identification number of minus joint

NUME = Number of members

NUPLJO = Identification number of plus joint

NULEDI = Number of left diagonal members

NURIDI = Number of right diagonal members

NUHOMB = Number of horizontal members

NUVEMB = Number of vertical members

NUVEJ1

NUVEJ2

NUVEJ3 Number of division points explained in DIPOHO

NUHOJ1

NUHOJ2

NUHOJ3

NLOTY = Number indicating kind of load

PRATIO = Poisson's ratio

POLOM1 = Point load magnitude

THICK = Thickness

VECOR = y-coord. of joint

VECOR1 = y-coord. of non suport joint

VELEN = Width of plate

VELEN1 = y-coordinate of lower edge of hole

VELEN2 = Width of hole

VELEN3 = VELEN - VELEN1 - VELEN2

WHY = y-coordinate of a point support

WHY2 = y-end coordinate of a line support

YOUNG = Young's modulus of elasticity

KILILO
NUPOLO = Extra variables not used by programs

APPENDIX F

DIRECTIONS TO USERS

I - Using the program as it is now

The following instructions are particularly useful in using the program as it is now. The input of course has to be given according to FORMATS already included in the program. Integers have to be right justified in their field. Floating point numbers can be anywhere in their field as long as the point is punched. The units to be used are KIPS and INCHES.

Necessary cards are:

Card 1 Format (5F 10.3)

In the first 10 columns put the span of the plate.

In columns 11-20 put the width of the plate.

In columns 21-30 put the thickness of the plate.

In columns 31-40 put Young's modulus.

In columns 41-50 put Poisson's ratio.

Card 2 Format (I6)

In columns 1-6, right justified put the desired maximum number of joints.

Card 3 Format (I4, 4F 10.3)

In column 4 put:

1 if there is no hole

2 if there is a hole

In case there is a hole additional information is necessary in this card. Namely:

Columns 5-14 Distance along span from the left edge of the plate to the left edge of hole.

Columns 15-24 Distance along width from the lower edge of the plate to the lower edge of hole.

Columns 25-34 Length of hole (in the direction of the span).

Columns 35-44 Width of hole (in the direction of the width of the plate).

Card 4 Format (2I5)

In columns 1-5 right justified, put the number of point supports.

In columns 6-10 right justified, put the number of line supports.

Card 5 Format (I4, 4F10.3)

There must be as many such cards as there are point supports.

In column 4 put a:

1 if support is hinged

2 if support is fixed

In columns 5-14 put the x-coordinate of particular point support. (See Fig. 4.1) (for location of axes)

In columns 5-14 put the y-coordinate of particular point support.

Card 5 Format (I4,2F10.3)

There must be as many such cards as there are line supports.

In column 4 put a:

1 if support is hinged

2 if support is fixed

In columns 5-14 put the x-coordinate of starting point of a line support. (In all the subsequent description the starting point of a line is the one to the left if the line is not parallel to the y-axis, or the one downwards if the line is parallel to the y-axis)

In columns 15-24 put the y-coordinate of the staring point of the line support.

In columns 25-34 put the x-coordinate of the end point of the line support.

In columns 35-44 put the y-coordinate of the end point of the line support.

Card 6 Format (3I4)

In columns 1-4 put the number of point loads, right justified.

In columns 5-8 put the number of line loads, right justified.

In columns 9-12 put the number of area loads, right justified.

Card 7 Format (I5,4F10.3)

In column 5 put a:

1 if load is a z force

2 if load is an x moment

3 if load is a y moment

(The system of coordinates is assumed right-handed and the loads are positive if their vector points towards the positive direction of the axes)

In columns 6-15 put the x-coordinate of the point of application of the load.

In columns 16-25 put the y-coordinate of the point of application of the load.

In columns 26-35 put the magnitude of the load.

There are as many such cards as there are point loads.

Card 6 Format (I5,3F10.3, I5, 3F10.3)

There are as many such cards as there are line loads.

In column 5 put a number (1,2 or 3) with the same meaning as in previous card.

In columns 6-15 put the x-coordinate of the starting point of the line load.

In columns 16-25 put the y-coordinate of the starting point of the line load.

In columns 26-35 put the magnitude of the load at the starting point.

In column 40 put a:

0 if line load is uniform

1 if line load is linear

In columns 41-50 put the x-coordinate of the end-point of the line load.

In columns 51-60 put the y-coordinate of the end-point of the line load.

In columns 61-70 put the magnitude of the load at the end point.

Card 7 Format (I5, 3F10.3, I5, 3F10.3)

There are as many such cards as there are area loads.

In column 5 put a number (1,2 or 3) with the same meaning as in previous card.

In columns 6-15 put the x coordinate of the left lower corner of the area load.

In columns 16-25 put the y coordinate of the left lower corner of the area load.

In columns 26-35 put the magnitude of the load at the starting point.

In column 40 put a:

0 if load is uniform

1 if load is linear along the x-direction

2 if load is linear along the y-direction

In columns 41-50 put the x-coordinate of the upper right corner.

In columns 51-60 put the y-coordinate of the upper right corner.

In columns 61-70 put the magnitude of the load at the end point.

Card 8 Format (2I4)

In columns 1-4 right justified put the number of different kinds of output desired.

Card 9 Format (I5 - 4F10.3)

There are as many such cards as there are kinds of output desired.

In column 5 put a:

2 if all reactions are desired

3 if all loads are desired at a particular point

4 if the loads are desired along a line

5 if all loads are desired

6 if displacements are desired at a particular point

7 if displacements are desired along a line

8 if all displacements are desired.

In case the above number is a 2,5 or 8 no other information is necessary on this card.

In case the above number is a 3 or a 6:

In columns 6-15 put the x-coordinate of the point where output is desired.

In columns 16-25 put the y-coordinate of the point where output is desired.

In case the above number is a 4 or a 7:

In columns 6-15 put the x-coordinate of the starting point of the line where output is desired.

In columns 16-25 put the y-coordinate of the starting point of the line where output is desired.

In columns 26-35 put the x-coordinate of the end point of the line where output is desired.

In columns 36-45 put the y-coordinate of the end point of the line where output is desired.

II Proposed use of the programs for CDL as in Appendix D

Following is a proposed user's formulation of a problem, if the form of CDL indicated in Appendix D, is used.

The statements have to be in the order indicated. The underlined part of a statement indicates that this is the way this statement is stored in the dictionary. Data identifiers can be omitted if the data follow in the specified ordere

Any LIST commands can appear one after the other.

```
SPAN ...
WIDTH ...
MAXIMUM NUMBER OF JOINTS ....
YOUNG'S MODULUS ....                                optional
POISSON'S RATIO ....
HOLE AT X ... AND Y.... SPAN ... WIDTH }        one or the
NO HOLE                                }          other

                            { FREE  }
POINT SUPPORTS    X ...  Y ...{ HINGED }        Free can be
                            { FIXED }           omitted
                  .      .
                  .      .
                  .      .
                  .      .
                  .      .
                  .      .
LINE SUPPORTS  X1 ... X2 ... Y1 ... Y2 ...      { FREE   }
                                                { HINGED }
                                                { FIXED  }
```

$$\underline{\text{POINT L}}\text{OADS}\quad \left\{\begin{array}{l}\underline{\text{Z FORCE}}\\ \underline{\text{X MOMENT}}\\ \underline{\text{Y MOMENT}}\end{array}\right\}\quad \underline{X}\ \ldots\ \underline{Y}\ \ldots\ \underline{P}\ \ldots$$

.

.

.

.

$$\underline{\text{LINE L}}\text{OADS}\quad \left\{\begin{array}{l}\underline{\text{FORCE Z LINEAR}}\\ \underline{\text{FORCE Z UNIFORM}}\\ \underline{\text{MOMENT X LINEAR}}\\ \underline{\text{MOMENT X UNIFORM}}\\ \underline{\text{MOMENT Y LINEAR}}\\ \underline{\text{MOMENT Y UNIFORM}}\end{array}\right\}\quad \underline{XA}..\underline{YA}...\underline{WA}...\underline{XB}...\underline{YB}...\underline{WB}...$$

.

.

$$\underline{\text{AREA L}}\text{OADS}\quad \left\{\begin{array}{l}\underline{\text{FORCE Z UNIFORM}}\\ \underline{\text{FORCE Z LINEAR ON X}}\\ \underline{\text{FORCE Z LINEAR ON Y}}\\ \underline{\text{MOMENT X UNIFORM}}\\ \underline{\text{MOMENT X ON X LINEAR}}\\ \underline{\text{MOMENT X ON Y LINEAR}}\\ \underline{\text{MOMENT Y UNIFORM}}\\ \underline{\text{MOMENT Y LINEAR ON X}}\\ \underline{\text{MOMENT Y LINEAR ON Y}}\end{array}\right\}\quad \underline{XA}..\underline{YA}..\underline{WA}..\underline{XB}..\underline{YB}..\underline{WB}..$$

$$\underline{\text{LIST}}\quad\left\{\begin{array}{l}\underline{\text{REACTIONS}}\\ \underline{\text{POINT LOADS}}\\ \underline{\text{LINE LOADS}}\\ \underline{\text{LOADS}}\\ \underline{\text{POINT DISPLACEMENTS}}\\ \underline{\text{LINE DISPLACEMENTS}}\\ \underline{\text{DISPLACEMENTS}}\end{array}\right\}\begin{array}{l}\\ \underline{\text{XA}}\ldots\ \underline{\text{YA}}\ \ldots\\ \underline{\text{XA}}\ \ldots\ \underline{\text{YA}}\ldots\ \underline{\text{XB}}\ldots\ \underline{\text{YB}}\ldots\\ \\ \underline{\text{XA}}\ldots\ \underline{\text{YA}}\ldots\\ \underline{\text{XA}}\ldots\ \underline{\text{YA}}\ldots\ \underline{\text{XB}}\ldots\ \underline{\text{YB}}\\ \\ \end{array}$$

thesK763
Computerized analysis of plates in flexu
3 2768 002 11713 7
DUDLEY KNOX LIBRARY